REJ09B0340-0200

16

M30245 Group

User's Manual

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER

M16C FAMILY / M16C/20 SERIES

User's Manual

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Technology Corp. without notice. Please review the latest information published by Renesas Technology Corp. through various means, including the Renesas Technology Corp. website (http://www.renesas.com).

Rev. 2.00
Revision date: Oct 16, 2006

RenesasTechnology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

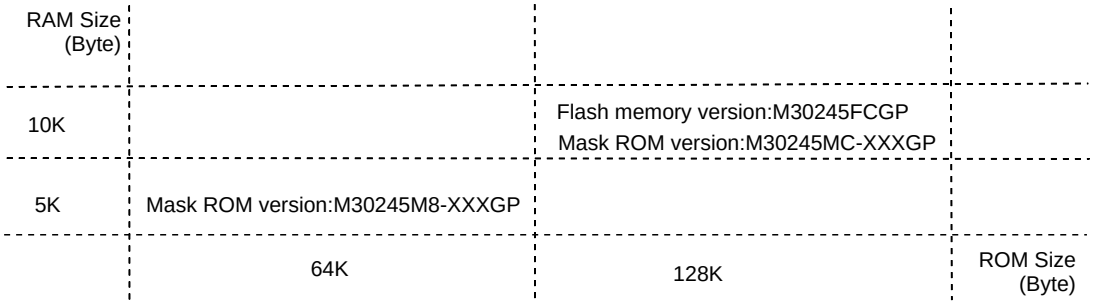
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

How to Use This Manual

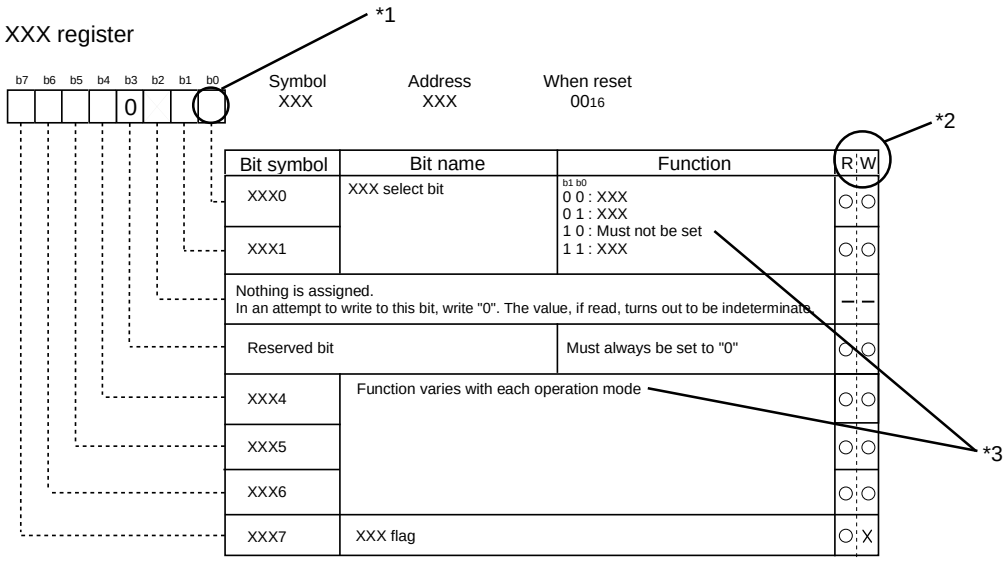
This user's manual is written for the M30245 group.
The reader of this manual is expected to have the basic knowledge of electric and logic circuits and microcomputers.
This manual explains a function of the following kind.

- M30245M8-XXXGP
- M30245MC-XXXGP
- M30245FCGP

These products have similar features except for the memories, which differ from one product to another. Be careful when writing a program, as the memories have different capacities.



The figure of each register configuration describes its functions and attributes as follows :



*1

Blank: Set to “0” or “1” according to intended use
0: Set to “0”
1: Set to “1”
X: Nothing is assigned

*2

R: Read
O.....Possible to read
X.....Impossible to read
–.....Nothing is assigned
W: Write
O.....Possible to write
X.....Written value is invalid
When write, value can be "0" or "1"
–.....Nothing is assigned

*3

Terms to use here are explained as follows.

- Nothing is assigned
  Nothing is assigned to the bit concerned. When write, set “0” for new function in future plan.
- Must not be set
  Not select. The operation at having selected is not guaranteed.
- Reserved bit
  Reserved bit. Set the specified value.
- Function varies with each operation mode
  Bit function changes according to the mode of peripheral functions.
- Always set to “0” in A mode.
  Set the corresponding bit to “0” in A mode.
- Invalid in A mode
  The bit concerned has no function in A mode. Set the specified value.
- Valid when bit A=“0”
  When bit A is “1”, the bit concerned has no function. When bit A is “0”, the bit concerned has function.

Table of Contents

Chapter 1. Hardware 1

Chapter 2. Peripheral Functions Usage 3

2.1 Protect 4
2.1.1 Overview 4
2.1.2 Protect Operation 5
2.2 Timer A 6
2.2.1 Overview 6
2.2.2 Operation of Timer A (timer mode) 12
2.2.3 Operation of Timer A (timer mode, gate function selected) 14
2.2.4 Operation of Timer A (timer mode, pulse output function selected) 16
2.2.5 Operation of Timer A (event counter mode, reload type selected) 18
2.2.6 Operation of Timer A (event counter mode, free run type selected) 20
2.2.7 Operation of timer A (two-phase pulse signal process in event counter mode, normal mode selected) 22
2.2.8 Operation of timer A (two-phase pulse signal process in event counter mode, multiply-by-4 mode selected) 24
2.2.9 Operation of Timer A (one-shot timer mode) 26
2.2.10 Operation of Timer A (pulse width modulation mode, 16-bit PWM mode selected) 28
2.2.11 Operation of Timer A (pulse width modulation mode, 8-bit PWM mode selected) 31
2.2.12 Precautions for Timer A (timer mode) 34
2.2.13 Precautions for Timer A (event counter mode) 35
2.2.14 Precautions for Timer A (one-shot timer mode) 37
2.2.15 Precautions for Timer A (pulse width modulation mode) 38
2.3 Clock-Synchronous Serial I/O 39
2.3.1 Overview 39
2.3.2 Operation of Serial I/O (transmission in clock-synchronous serial I/O mode) 45
2.3.3 Operation of Serial I/O (reception in clock-synchronous serial I/O mode) 49
2.3.4 Precautions for Serial I/O (in clock-synchronous serial I/O mode) 53
2.4 Clock-Asynchronous Serial I/O (UART) 55
2.4.1 Overview 55
2.4.2 Operation of Serial I/O (transmission in UART mode) 64
2.4.3 Operation of Serial I/O (reception in UART mode) 68
2.4.4 Serial I/O Precautions (UART Mode) 72
2.4.5 Operation of Serial I/O (transmission used for SIM interface) 73
2.4.6 Operation of Serial I/O (reception used for SIM interface) 77
2.4.7 Clock Signals in used for the SIM Interface 81
2.5 Serial Interface Special Function 85
2.5.1 Overview 85
2.5.2 Operation of Serial Interface Special Function (transmission in master mode without delay) 94

2.5.3 Operation of Serial Interface Special Function (reception in master mode with clock delay) 98
2.5.4 Operation of Serial Interface Special Function (transmission in slave mode without delay) 102
2.5.5 Operation of Serial Interface Special Function (reception in slave mode with clock delay) 106
2.6 Serial sound interface 110
2.6.1 Overview 110
2.6.2 Example of Serial Sound Interface operation 116
2.6.3 Precautions for Serial Sound Interface 120
2.7 Frequency synthesizer (PLL) 121
2.7.1 Overview 121
2.7.2 Operation of frequency synthesizer 124
2.7.3 Precautions for Frequency synthesizer 126
2.8 USB function 127
2.8.1 Overview 127
2.8.2 USB function control 139
2.8.3 USB Interrupt 150
2.8.4 USB Operation (Suspend/Resume Function) 161
2.8.5 USB Operation (Endpoint 0) 169
2.8.6 USB Operation (Endpoints 1 to 4 Receive) 182
2.8.7 USB Operation (Endpoints 1 to 4 Transmit) 193
2.8.8 USB Operation (Interface with DMAC Transfer) 207
2.8.9 Precautions for USB 210
2.9 A/D Converter 213
2.9.1 Overview 213
2.9.2 Operation of A/D converter (one-shot mode) 218
2.9.3 Operation of A/D Converter (in one-shot mode, an external trigger selected) 220
2.9.4 Operation of A/D Converter (in repeat mode) 222
2.9.5 Operation of A/D Converter (in single sweep mode) 224
2.9.6 Operation of A/D Converter (in repeat sweep mode 0) 226
2.9.7 Operation of A/D Converter (in repeat sweep mode 1) 228
2.9.8 Precautions for A/D Converter 230
2.9.9 Method of A/D Conversion (10-bit mode) 231
2.9.10 Method of A/D Conversion (8-bit mode) 233
2.9.11 Absolute Accuracy and Differential Non-Linearity Error 235
2.9.12 Internal Equivalent Circuit of Analog Input 237
2.9.13 Sensor's Output Impedance under A/D Conversion (reference value) 238
2.10 DMAC Usage 240
2.10.1 Overview of the DMAC usage 240
2.10.2 Operation of DMAC (one-shot transfer mode) 245
2.10.3 Operation of DMAC (repeated transfer mode) 247
2.11 CRC Calculation Circuit 249
2.11.1 Overview 249
2.11.2 Operation of CRC Calculation Circuit 251
2.11.3 SFR Access Snoop Function 252
2.12 Watchdog Timer 253

2.12.1 Overview 253
2.12.2 Operation of Watchdog Timer (Watchdog timer interrupt) 256
2.13 Address Match Interrupt Usage 258
2.13.1 Overview of the address match interrupt usage 258
2.13.2 Operation of Address Match Interrupt 260
2.14 Key-Input Interrupt Usage 262
2.14.1 Overview of the key-input interrupt usage 262
2.14.2 Operation of Key-Input Interrupt 265
2.15 Multiple interrupts Usage 267
2.15.1 Overview of the Multiple interrupts usage 267
2.15.2 Multiple Interrupts Operation 272
2.16 Power Control Usage 274
2.16.1 Overview of the power control usage 274
2.16.2 Stop Mode Set-Up 280
2.16.3 Wait Mode Set-Up 281
2.16.4 Precautions in Power Control 282
2.17 Programmable I/O Ports Usage 284
2.17.1 Overview of the programmable I/O ports usage 284

Chapter 3. Examples of Peripheral Functions Applications 293
3.1 Long-Period Timers 295
3.2 Variable-Period Variable-Duty PWM Output 299
3.3 Buzzer Output 303
3.4 Solution for External Interrupt Pins Shortage 305
3.5 Memory to Memory DMA Transfer 307
3.7 Buzzer Output 311
3.6 CRC Calculation SFR Access Snoop Function in Clock Synchronous Serial Data Transmit 311
3.7 Transfer from USB FIFO to Serial Sound Interface 316
3.8 Controlling Power Using Stop Mode 321
3.9 Controlling Power Using Wait Mode 325

Chapter 4. External Buses 329
4.1 Overview of External Buses 330
4.2 Data Access 331
4.2.1 Data Bus Width 331
4.2.2 Chip Selects and Address Bus 332
4.2.3 R/W Modes 333
4.3 Connection Examples 334
4.3.1 16-bit Memory to 16-bit Width Data Bus Connection Example 334
4.3.2 8-bit Memory to 16-bit Width Data Bus Connection Example 335

4.3.3 8-bit Memory to 8-bit Width Data Bus Connection Example 337
4.3.4 Two 8-bit and 16-Bit Memory to 16-Bit Width Data Bus Connection Example 338
4.3.5 Chip Selects and Address Bus 339
4.4 Connectable Memories 340
4.4.1 Operation Frequency and Access Time 340
4.4.2 Connecting Low-Speed Memory 343
4.4.3 Connectable Memories 346
4.5 Releasing an External Bus ($\overline{\text{HOLD}}$ input and $\overline{\text{HLDA}}$ output) 347
4.6 Precautions for External Bus 349

Chapter 5. Standard Characteristics 351
5.1 DC Standard Characteristics 352
5.1.1 Port Standard Characteristics 352
5.1.2 VCC-ICC Characteristics 354

Chapter 1

Hardware

See M30245 group datasheet.

Chapter 2

Peripheral Functions Usage

2.1 Protect

2.1.1 Overview

'Protect' is a function that causes a value held in a register to be unchanged even when a program runs away. The following is an overview of the protect function:

(1) Registers affected by the protect function

The registers affected by the protect function are:

(a) System clock control registers 0, 1 (addresses 0006_{16} and 0007_{16})

(b) Processor mode registers 0, 1 (addresses 0004_{16} and 0005_{16})

(c) Frequency synthesizer-related registers (address $03DB_{16}$ to $03DF_{16}$)

The values in registers (a) through (c) cannot be changed in write-protect state. To change values in the registers, put the individual registers in write-enabled state.

(2) Protect register

Figure 2.1.1 shows protect register.

Protect register

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	X	0		

Symbol: PRCR
Address: $000A_{16}$
When reset: $XXXXX000_2$

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16}) and frequency synthesizer registers (addresses $03DB_{16}$ to $03DF_{16}$)	0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
Reserved		Must always be set to “0”	○	○
Nothing is assigned. When write, set “0”. When read, their contents are indeterminate.			–	–

Figure 2.1.1. Protect register

2.1.2 Protect Operation

The following explains the protect operation. Figure 2.1.2 shows the set-up procedure.

Operation (1) Setting “1” in the write-enable bit of system clock control registers 0 and 1 and frequency synthesizer-related registers causes system clock control register 0 and 1 and frequency synthesizer-related registers to be in write-enabled state.

(2) The contents of system clock control register 0 and 1 and these of frequency synthesizer-related registers are changed.

(3) Setting “0” in PRC0 causes system clock control register 0 and 1 and frequency synthesizer-related registers to be in write-inhibited state.

(4) To change the contents of processor mode register 0 and that of processor mode register 1, follow the same steps as in dealing with system clock control registers and frequency synthesizer-related registers.

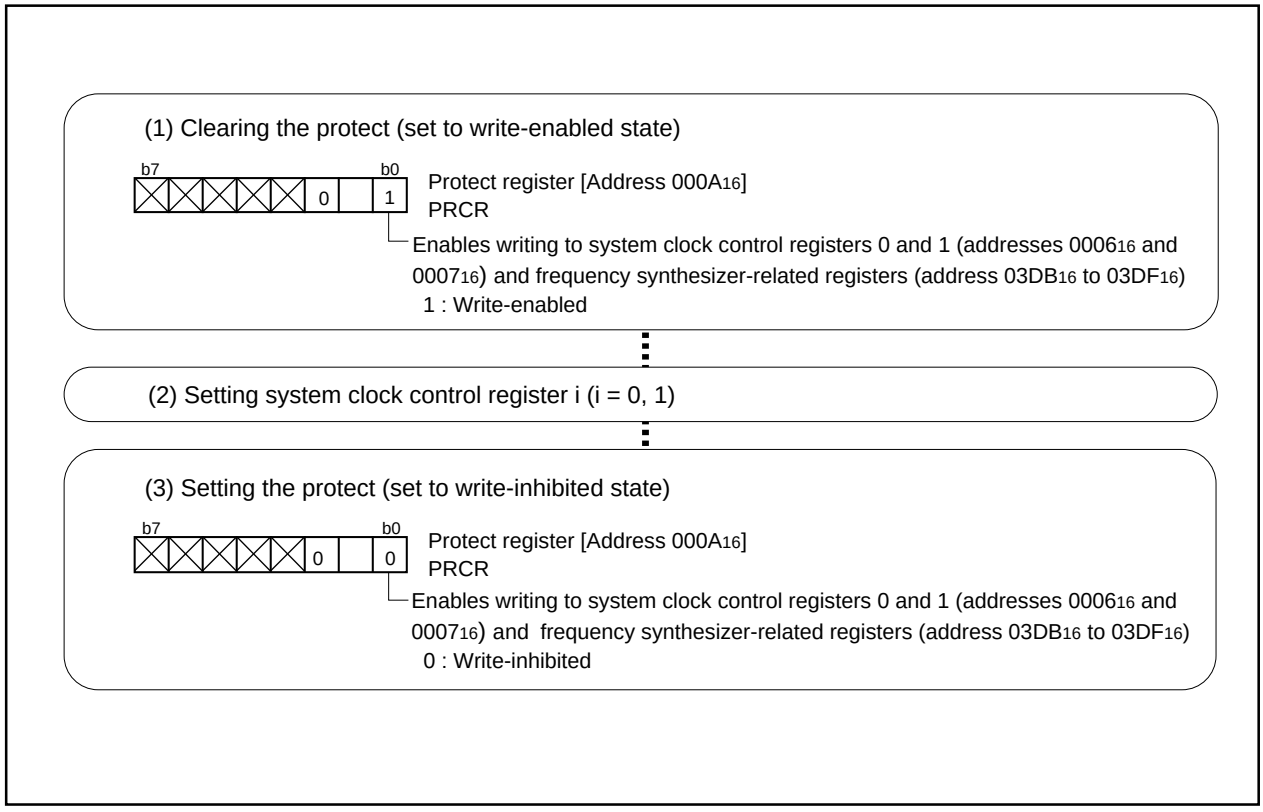

(1) Clearing the protect (set to write-enabled state)
b7
b0
0
1
Protect register [Address 000A16]
PRCR
Enables writing to system clock control registers 0 and 1 (addresses 000616 and 000716) and frequency synthesizer-related registers (address 03DB16 to 03DF16)
1 : Write-enabled
(2) Setting system clock control register i (i = 0, 1)
(3) Setting the protect (set to write-inhibited state)
b7
b0
0
0
Protect register [Address 000A16]
PRCR
Enables writing to system clock control registers 0 and 1 (addresses 000616 and 000716) and frequency synthesizer-related registers (address 03DB16 to 03DF16)
0 : Write-inhibited

Figure 2.1.2. Set-up procedure for protect function

2.2 Timer A

2.2.1 Overview

The following is an overview for timer A, a 16-bit timer.

(1) Mode

Timer A operates in one of the four modes:

(a) Timer mode

In this mode, the internal count source is counted. Two functions can be selected: the pulse output function that reverses output from a port every time an overflow occurs, or the gate function which controls the count start/stop according to the input signal from a port.

(b) Event counter mode

This mode counts the pulses from the outside and the number of overflows in other timers. The free-run type, in which nothing is reloaded from the reload register, can be selected when an underflow occurs. The pulse output function can also be selected. Please refer to the timer mode explanation for details, as the operation is identical.

Furthermore, Timer A has a two-phase pulse signal processing function which generates an up count or down count in the event counter mode, depending on the phase of the two input signals. The normal mode or 4-multiplication mode can be selected depending on the phase detective method.

(c) One-shot timer mode

In this mode, the timer is started by the trigger and stops when the timer goes to “0”. The trigger can be selected from the following 2 types: an overflow of the timer, or a software trigger. The pulse output function can also be selected. Please refer to the timer mode explanation for details, as the operation is identical.

(d) Pulse width modulation (PWM) mode

In this mode, the arbitrary pulses are successively output. Either a 16-bit fixed-period PWM mode or 8-bit variable-period mode can be selected. The trigger for initiating output can also be selected. Please refer to the one-shot timer mode explanation for details, as the operation is identical.

(2) Count source

The internal count source can be selected from f_1, f_8, f_{32}, and f_{C32}. Clocks f_1, f_8, and f_{32} are derived by dividing the CPU's main clock by 1, 8, and 32 respectively. Clock f_{C32} is derived by dividing the CPU's secondary clock by 32.

(3) Frequency division ratio

In timer mode or pulse width modulation mode, [the value set in the timer register + 1] becomes the frequency division ratio. In event counter mode, [the set value + 1] becomes the frequency division ratio when a down count is performed, or [$FFFF_{16}$ - the set value + 1] becomes the frequency division ratio when an up count is performed. In one-shot timer mode, the value set in the timer register becomes the frequency division ratio.

The counter overflows (or underflows) when a count source equal to a frequency division ratio is input, and an interrupt occurs. For the pulse output function, the output from the port varies (the value in the port register does not vary).

(4) Reading the timer

Either in timer mode or in event counter mode, reading the timer register takes out the count at that moment. Read it in 16-bit units. The data either in one-shot timer mode or in pulse width modulation mode is indeterminate.

(5) Writing to the timer

To write to the timer register when a count is in progress, the value is written only to the reload register. When writing to the timer register when a count is stopped, the value is written both to the reload register and to the counter. Write a value in 16-bit units.

(6) Relation between the input/output to/from the timer and the direction register

With the output function of the timer, pulses are output regardless of the contents of the port direction register. To input an external signal to the timer, set the port direction register to input.

(7) Pins related to timer A

(a) $TA0_{IN}$, $TA1_{IN}$, $TA2_{IN}$, $TA3_{IN}$, $TA4_{IN}$	Input pins to timer A.
(b) $TA0_{OUT}$, $TA1_{OUT}$, $TA2_{OUT}$, $TA3_{OUT}$, $TA4_{OUT}$	Output pins from timer A. They become input pins to timer A when event counter mode is active.

(8) Registers related to timer A

Figure 2.2.1 shows the memory map of timer A-related registers. Figures 2.2.2 through 2.2.5 show timer A-related registers.

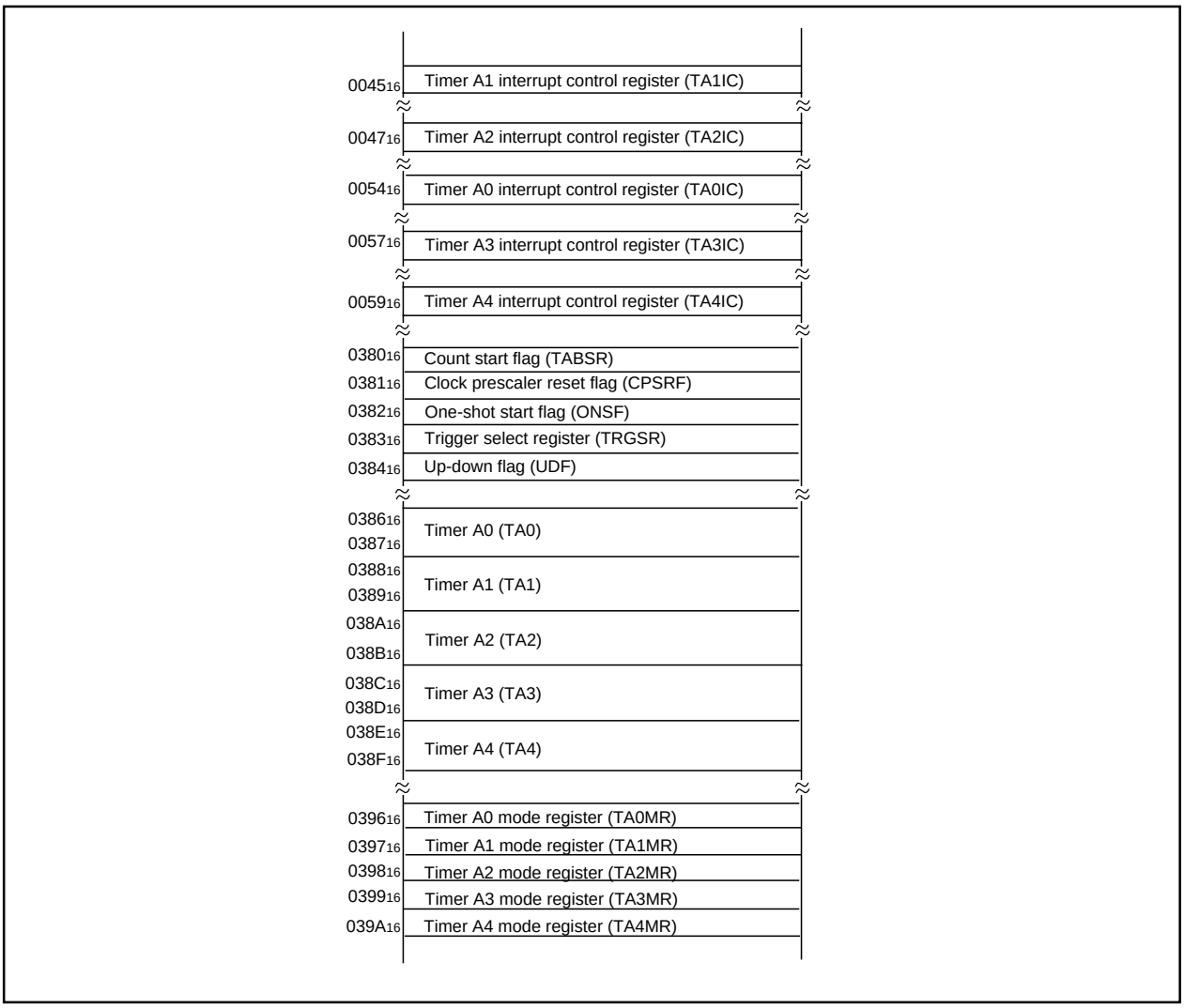

0045₁₆ Timer A1 interrupt control register (TA1IC)
0047₁₆ Timer A2 interrupt control register (TA2IC)
0054₁₆ Timer A0 interrupt control register (TA0IC)
0057₁₆ Timer A3 interrupt control register (TA3IC)
0059₁₆ Timer A4 interrupt control register (TA4IC)
0380₁₆ Count start flag (TABSR)
0381₁₆ Clock prescaler reset flag (CPSRF)
0382₁₆ One-shot start flag (ONSF)
0383₁₆ Trigger select register (TRGSR)
0384₁₆ Up-down flag (UDF)
0386₁₆ 0387₁₆ Timer A0 (TA0)
0388₁₆ 0389₁₆ Timer A1 (TA1)
038A₁₆ 038B₁₆ Timer A2 (TA2)
038C₁₆ 038D₁₆ Timer A3 (TA3)
038E₁₆ 038F₁₆ Timer A4 (TA4)
0396₁₆ Timer A0 mode register (TA0MR)
0397₁₆ Timer A1 mode register (TA1MR)
0398₁₆ Timer A2 mode register (TA2MR)
0399₁₆ Timer A3 mode register (TA3MR)
039A₁₆ Timer A4 mode register (TA4MR)

Figure 2.2.1. Memory map of timer A-related registers

Timer Ai mode register (i = 0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: TAiMR (i=0 to 4)

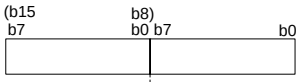
Address: 0396₁₆ to 039A₁₆

When reset: 00₁₆

Bit Symbol	Bit Name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : PWM mode	O	O
TMOD1			O	O
MR0	———	Function varies with each mode operation	O	O
MR1			O	O
MR2			O	O
MR3			O	O
TCK0	Count source select bit	Function varies with each mode operation	O	O
TCK1			O	O

Figure 2.2.2. Timer A-related registers (1)

Timer Ai register (i = 0 to 4) (Note 1)

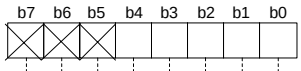


Symbol	Address	When reset
TA0	0387₁₆, 0386₁₆	Indeterminate
TA1	0389₁₆, 0388₁₆	Indeterminate
TA2	038B₁₆, 038A₁₆	Indeterminate
TA3	038D₁₆, 038C₁₆	Indeterminate
TA4	038F₁₆, 038E₁₆	Indeterminate

Mode	Function	Values that can be set	R	W
Timer mode	16-bit counter (set to divide ratio)	0000₁₆ to FFFF₁₆	O	O
Event counter mode	16-bit counter (set to divide ratio) (Note 2)	0000₁₆ to FFFF₁₆	O	O
One-shot timer mode	16-bit counter (set to one-shot width) (Note 6)	0000₁₆ to FFFF₁₆ (Note 3)	X	O
16-bit PWM	16-bit PWM (set to PWM pulse "H" width) (Note 4, 7)	0000₁₆ to FFFF₁₆ (Note 3)	X	O
8-bit PWM	Low-order bits: 8-bit prescaler (set to PWM period) (Notes 5, 7) High-order bits : 8-bit PWM (set to PWM pulse "H" width) (Notes 5, 7)	00₁₆ to FE₁₆ (Both high-order and low-order addresses) (Note 3)	X	O

Note 1 : Read and write data in 16-bit units.
Note 2 : Counts pulses from an external source of timer overflow.
Note 3 : Use MOV instruction to write to this register.
Note 4 : When setting value is n, PWM period and “H” width of PWM pulses are:
PWM period : $(2^{16}- 1)/fi$
PWM pulse “H” width : n/fi
Note 5 : When setting value of high-order address is n and setting value of low-order address is m, PWM period and “H” width of PWM pulse are:
PWM period : $(2^{8-} 1)$ X (m + 1)/fi
PWM pulse “H” width : (m + 1)n/fi
Note 6 : When the Timer Ai register is set to “0000₁₆”, the counter does not operate and the Timer Ai interrupt request is not generated. When the pulse is se to output, the pulse does not output from the TAiOUT pin.
Note 7 : When the Timer Ai register is set to “0000₁₆”, the pulse width modulator does not operate and the output level of the TAiOUT pin remains “L” level, therefore the Timer Ai interrupt request is not generated. This also occurs in the 8-bit pulse width modulator mode when the significant 8 high-order bits in the Timer Ai register are set to “00₁₆”

Count start flag



Symbol	Address	When reset
TABSR	0380₁₆	XXX00000₂

Bit Symbol	Bit Name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	O	O
TA1S	Timer A1 count start flag		O	O
TA2S	Timer A2 count start flag		O	O
TA3S	Timer A3 count start flag		O	O
TA4S	Timer A4 count start flag		O	O
Nothing is assigned. Write “0” when writing to these bits. The contents are indeterminate if read.			–	–

Figure 2.2.3. Timer A-related registers (2)

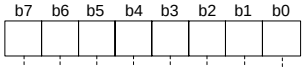
Up/down flag (Note)

Symbol: UDF | Address: 038416 | When reset: 0016

Bit Symbol	Bit Name	Function	R	W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count This specification becomes valid when the up/down flag content is selected for up/down switching cause	O	O
TA1UD	Timer A1 up/down flag		O	O
TA2UD	Timer A2 up/down flag		O	O
TA3UD	Timer A3 up/down flag		O	O
TA4UD	Timer A4 up/down flag		O	O
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : Disabled 1 : Enabled When not using the two-phase pulse signal processing function, set the select bit to “0”	–	O
TA3P	Timer A3 two-phase pulse signal processing select bit		–	O
TA4P	Timer A4 two-phase pulse signal processing select bit		–	O

Note : Use MOV instruction to write to this register

Trigger select register



Symbol: TRGSR | Address: 038316 | When reset: 0016

Bit Symbol	Bit Name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on TA1IN is selected (Note) 0 1 : Invalid 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	O	O
TA1TGH			O	O
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on TA2IN is selected (Note) 0 1 : Invalid 1 0 : TA1 overflow is selected 1 1 : TA3 overflow is selected	O	O
TA2TGH			O	O
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on TA3IN is selected (Note) 0 1 : Invalid 1 0 : TA2 overflow is selected 1 1 : TA4 overflow is selected	O	O
TA3TGH			O	O
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Input on TA4IN is selected (Note) 0 1 : Invalid 1 0 : TA3 overflow is selected 1 1 : TA0 overflow is selected	O	O
TA4TGH			O	O

Note: Set the corresponding port direction register to “0”.

Figure 2.2.4. Timer A-related registers (3)

Clock prescaler reset flag

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$

Bit Symbol	Bit Name	Function	R	W
Nothing is assigned. Write "0" when writing to these bits. The contents are indeterminate if read.			–	–
CPSR	Clock prescaler reset flag	0 : No effect 1 : Reset (The value is “0” when read)	O	O

One-shot start flag

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol	Address	When reset
ONSF	0382_{16}	00_{16}

Bit Symbol	Bit Name	Function	R	W
TA0OS	Timer A0 one-shot start flag	0 : Invalid 1 : Timer start (Note 1)	O	O
TA1OS	Timer A1 one-shot start flag		O	O
TA2OS	Timer A2 one-shot start flag		O	O
TA3OS	Timer A3 one-shot start flag		O	O
TA4OS	Timer A4 one-shot start flag		O	O
Reserved		Always set to “0”	O	O
TA0TGL	Timer A0 event/trigger select bit	b1 b0 0 0 : Input on TA0IN is selected (Notes 2, 3) 0 1 : Invalid 1 0 : TA4 overflow is selected 1 1 : TA1 overflow is selected	O	O
TA0TGH			O	O

Note 1 : The value is “0” when read.
Note 2 : Set the corresponding pin's port direction register to “0”.
Note 3 : To start count in one-shot timer mode, do not use an extrenal trigger input.

Figure 2.2.5. Timer A-related registers (4)

2.2.2 Operation of Timer A (timer mode)

In timer mode, choose functions from those listed in Table 2.2.1. Operations of the circled items are described below. Figure 2.2.6 shows the operation timing, and Figure 2.2.7 shows the set-up procedure.

Table 2.2.1. Choosed functions

Item	Set-up	
Count source	**O**	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
Pulse output function	**O**	No pulses output
		Pulses output
Gate function	**O**	No gate function
		Performs count only for the period in which the TAi_{IN} pin is at "L" level
		Performs count only for the period in which the TAi_{IN} pin is at "H" level

Operation (1) Setting the count start flag to "1" causes the counter to perform a down count on the count source.
(2) If an underflow occurs, the content of the reload register is reloaded, and the count continues. At this time, the timer Ai interrupt request bit goes to "1".
(3) Setting the count start flag to "0" causes the counter to hold its value and to stop.

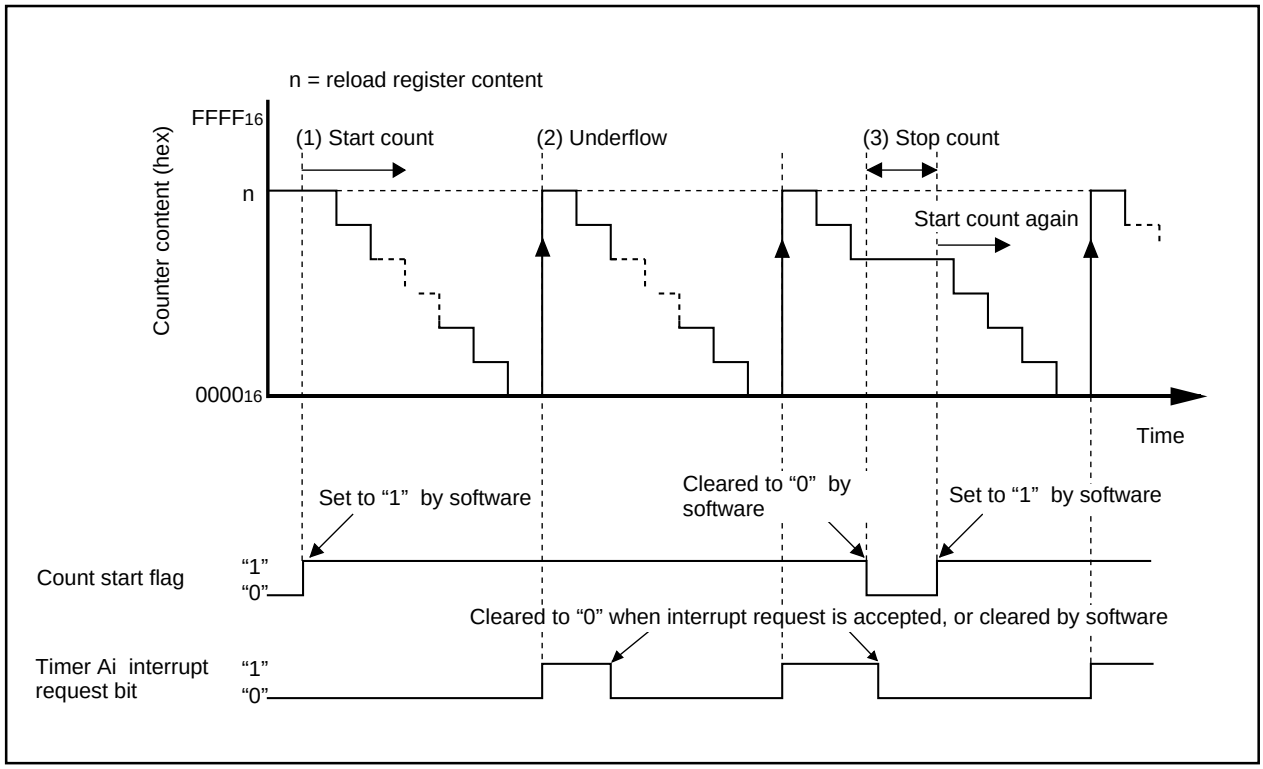


Figure 2.2.6. Operation timing of timer mode

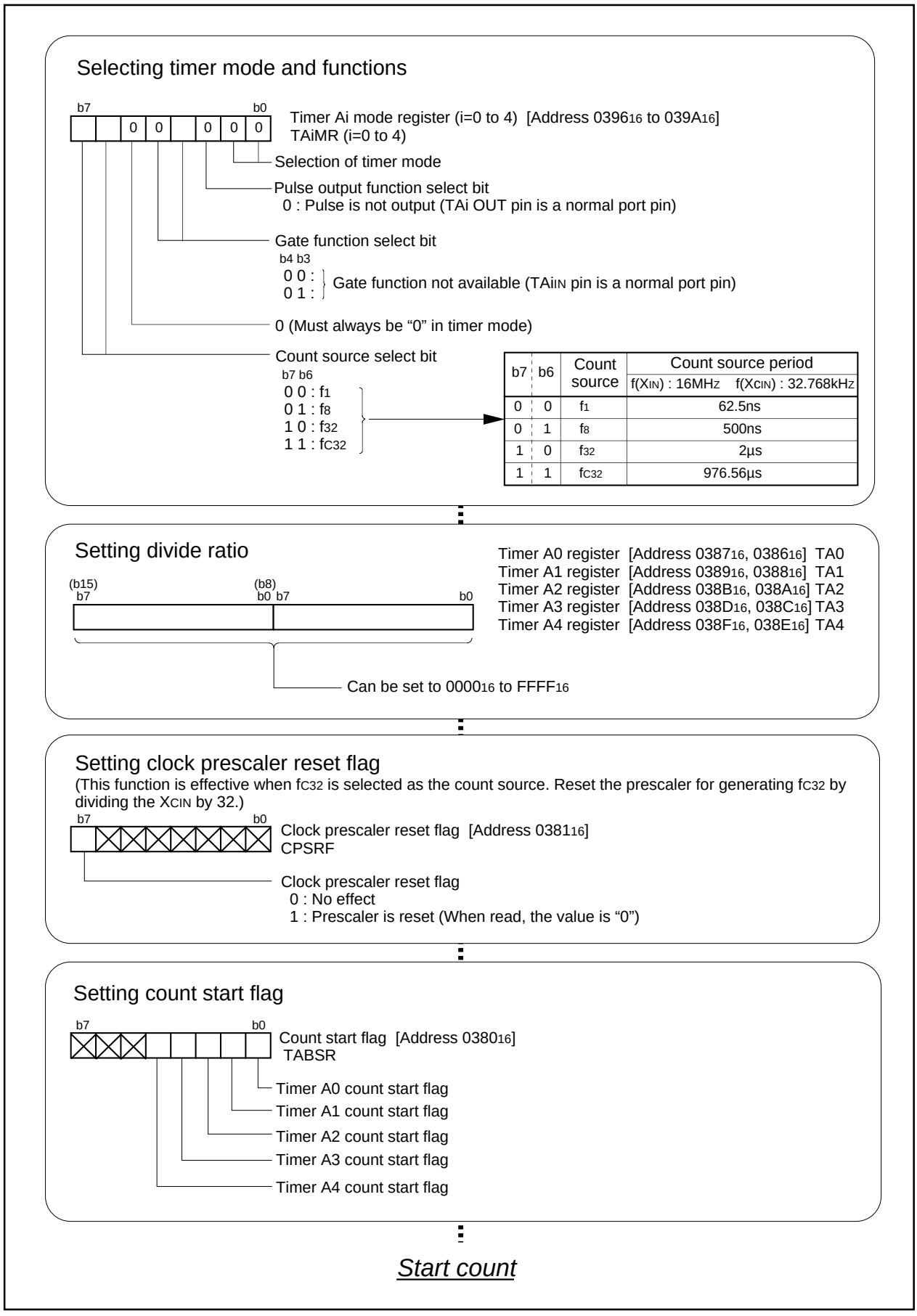

Selecting timer mode and functions
b7 b0
0 0 0 0 0
Timer Ai mode register (i=0 to 4) [Address 039616 to 039A16]
TAiMR (i=0 to 4)
Selection of timer mode
Pulse output function select bit
0 : Pulse is not output (TAi OUT pin is a normal port pin)
Gate function select bit
b4 b3
0 0 :
0 1 : Gate function not available (TAiIN pin is a normal port pin)
0 (Must always be "0" in timer mode)
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fC32
b7 b6 Count source Count source period
f(XIN) : 16MHz f(XCIN) : 32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fC32 976.56µs
Setting divide ratio
(b15) (b8)
b7 b0 b7 b0
Timer A0 register [Address 038716, 038616] TA0
Timer A1 register [Address 038916, 038816] TA1
Timer A2 register [Address 038B16, 038A16] TA2
Timer A3 register [Address 038D16, 038C16] TA3
Timer A4 register [Address 038F16, 038E16] TA4
Can be set to 000016 to FFFF16
Setting clock prescaler reset flag
(This function is effective when fC32 is selected as the count source. Reset the prescaler for generating fC32 by dividing the XCIN by 32.)
b7 b0
Clock prescaler reset flag [Address 038116]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")
Setting count start flag
b7 b0
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
Timer A1 count start flag
Timer A2 count start flag
Timer A3 count start flag
Timer A4 count start flag
Start count

Figure 2.2.7. Set-up procedure of timer mode

2.2.3 Operation of Timer A (timer mode, gate function selected)

In timer mode, choose functions from those listed in Table 2.2.2. Operations of the circled items are described below. Figure 2.2.8 shows the operation timing, and Figure 2.2.9 shows the set-up procedure.

Table 2.2.2. Choosed functions

Item	Set-up	
Count source	**O**	Internal count source(f_1 / f_8 / f_{32} / fc_{32})
Pulse output function	**O**	No pulses output
		Pulses output
Gate function		No gate function
		Performs count only for the period in which the TAiIN pin is at "L" level
	O	Performs count only for the period in which the TAiIN pin is at "H" level

Operation
(1) When the count start flag is set to "1" and the TAiIN pin inputs at "H" level, the counter performs a down count on the count source.
(2) When the TAiIN pin inputs at "L" level, the counter holds its value and stops.
(3) If an underflow occurs, the content of the reload register is reloaded and the count continues. At this time, the timer Ai interrupt request bit goes to "1".
(4) Setting the count start flag to "0" causes the counter to hold its value and to stop.

Note
• Make the pulse width of the signal input to the TAiIN pin not less than two cycles of the count source.

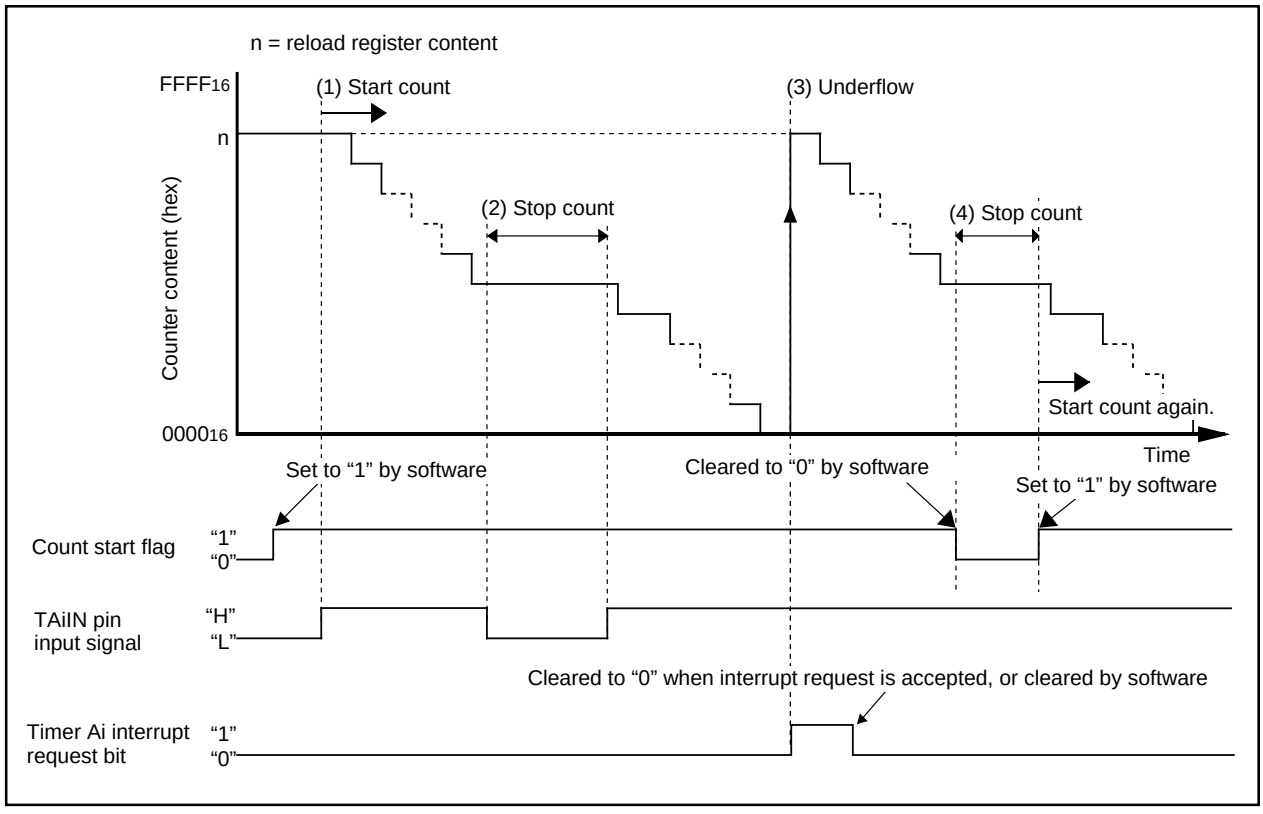


Figure 2.2.8. Operation timing of timer mode, gate function selected

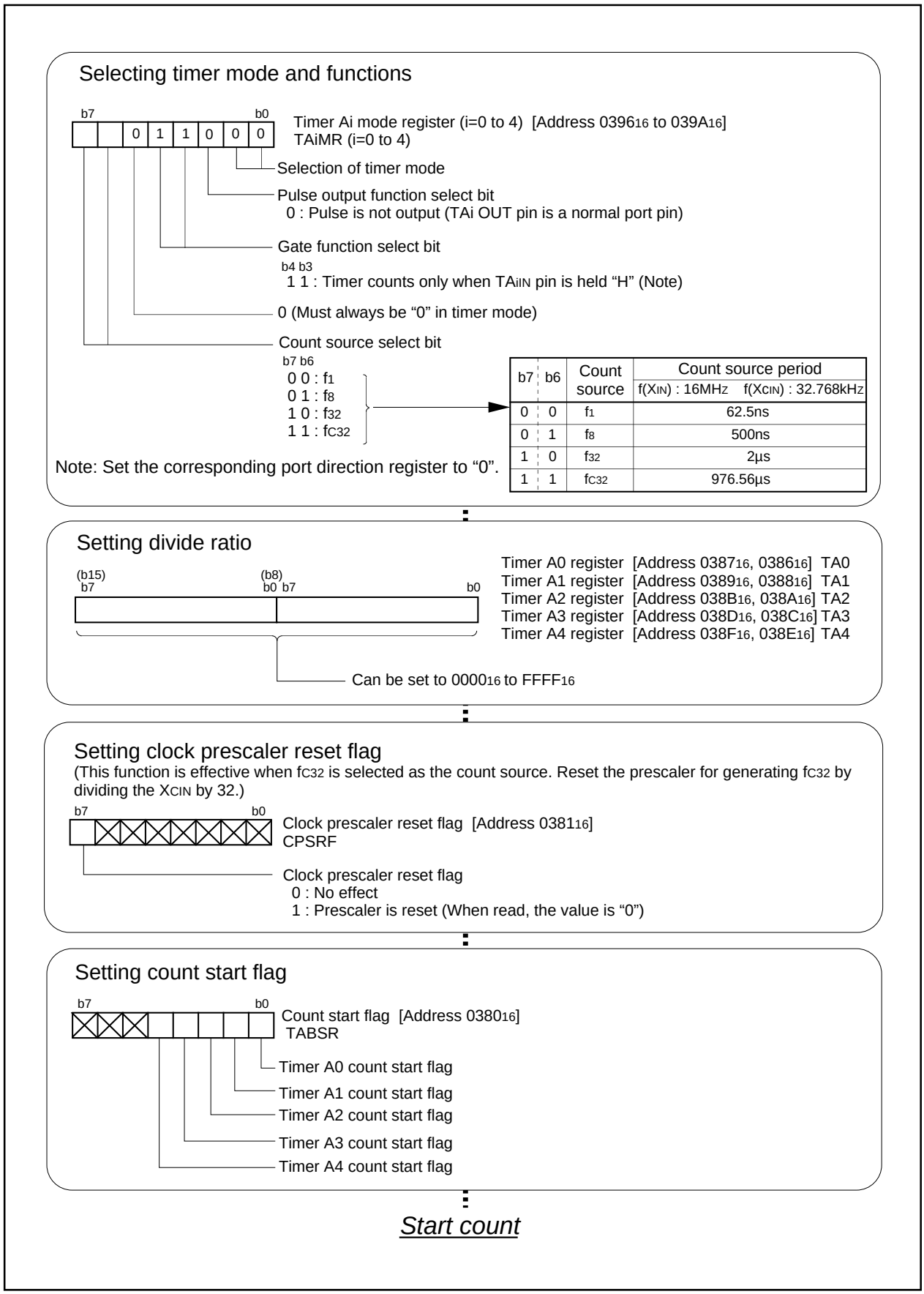

Selecting timer mode and functions
b7 b0
0 1 1 0 0 0
Timer Ai mode register (i=0 to 4) [Address 039616 to 039A16]
TAiMR (i=0 to 4)
Selection of timer mode
Pulse output function select bit
0 : Pulse is not output (TAi OUT pin is a normal port pin)
Gate function select bit
b4 b3
1 1 : Timer counts only when TAiIN pin is held "H" (Note)
0 (Must always be "0" in timer mode)
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fc32
b7 b6 Count source Count source period
f(XIN) : 16MHz f(XCIN) : 32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fc32 976.56µs
Note: Set the corresponding port direction register to "0".
Setting divide ratio
(b15) b7 (b8) b0 b7 b0
Timer A0 register [Address 038716, 038616] TA0
Timer A1 register [Address 038916, 038816] TA1
Timer A2 register [Address 038B16, 038A16] TA2
Timer A3 register [Address 038D16, 038C16] TA3
Timer A4 register [Address 038F16, 038E16] TA4
Can be set to 000016 to FFFF16
Setting clock prescaler reset flag
(This function is effective when fc32 is selected as the count source. Reset the prescaler for generating fc32 by dividing the XCIN by 32.)
b7 b0
Clock prescaler reset flag [Address 038116]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")
Setting count start flag
b7 b0
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
Timer A1 count start flag
Timer A2 count start flag
Timer A3 count start flag
Timer A4 count start flag
Start count

Figure 2.2.9. Set-up procedure of timer mode, gate function selected

2.2.4 Operation of Timer A (timer mode, pulse output function selected)

In timer mode, choose functions from those listed in Table 2.2.3. Operations of the circled items are described below. Figure 2.2.10 shows the operation timing, and Figures 2.2.11 shows the set-up procedure.

Table 2.2.3. Choosed functions

Item	Set-up	
Count source	O	Internal count source(f_1 / f_8 / f_{32} / fc_{32})
Pulse output function		No pulses output
	O	Pulses output
Gate function	O	No gate function
		Performs count only for the period in which the TAi_{IN} pin is at "L" level
		Performs count only for the period in which the TAi_{IN} pin is at "H" level

Operation (1) Setting the count start flag to "1" causes the counter to perform a down count on the count source.

(2) If an underflow occurs, the content of the reload register is reloaded and the count continues. At this time, the timer Ai interrupt request bit goes to "1". Also, the output polarity of the TAi_{OUT} pin reverses.

(3) Setting the count start flag to "0" causes the counter to hold its value and to stop. Also, the TAi_{OUT} pin outputs an "L" level.

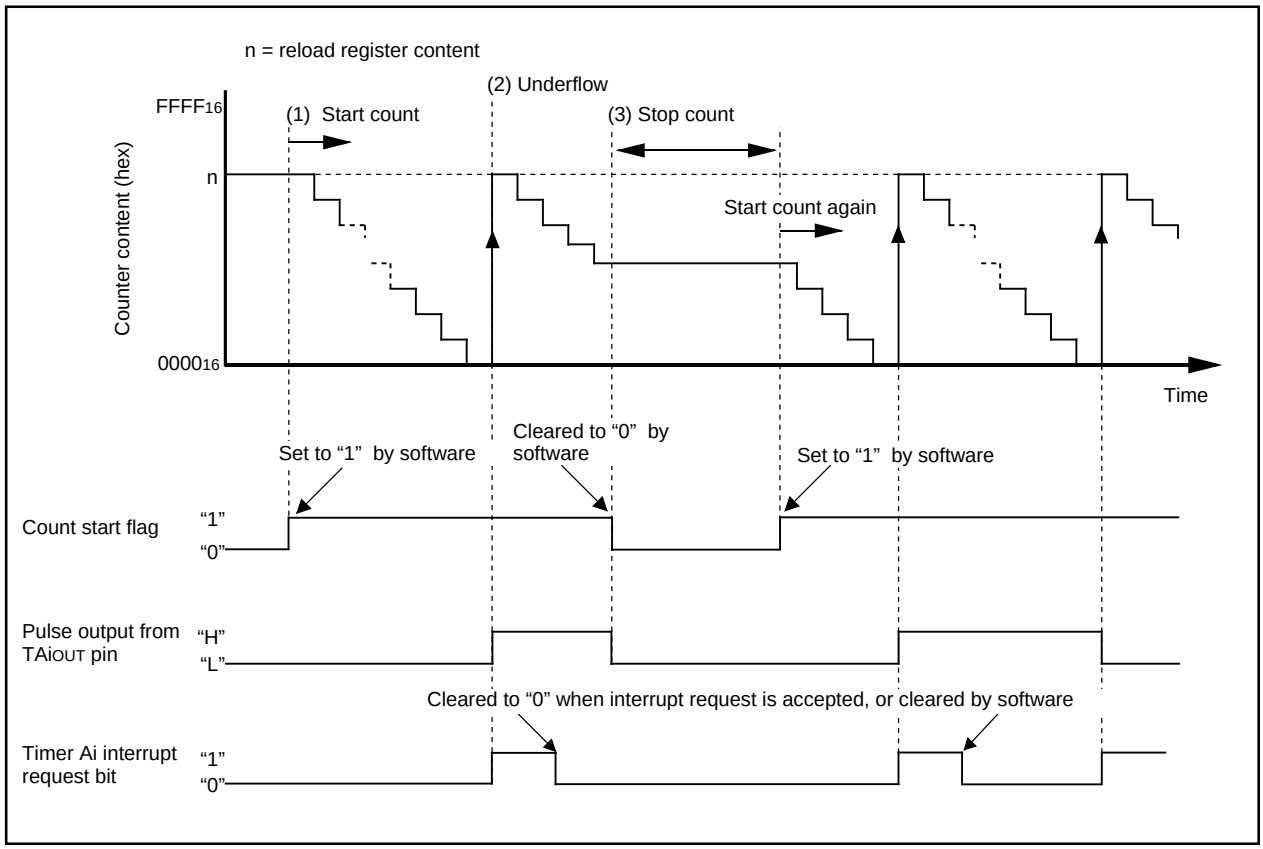


Figure 2.2.10. Operation timing of timer mode, pulse output function selected

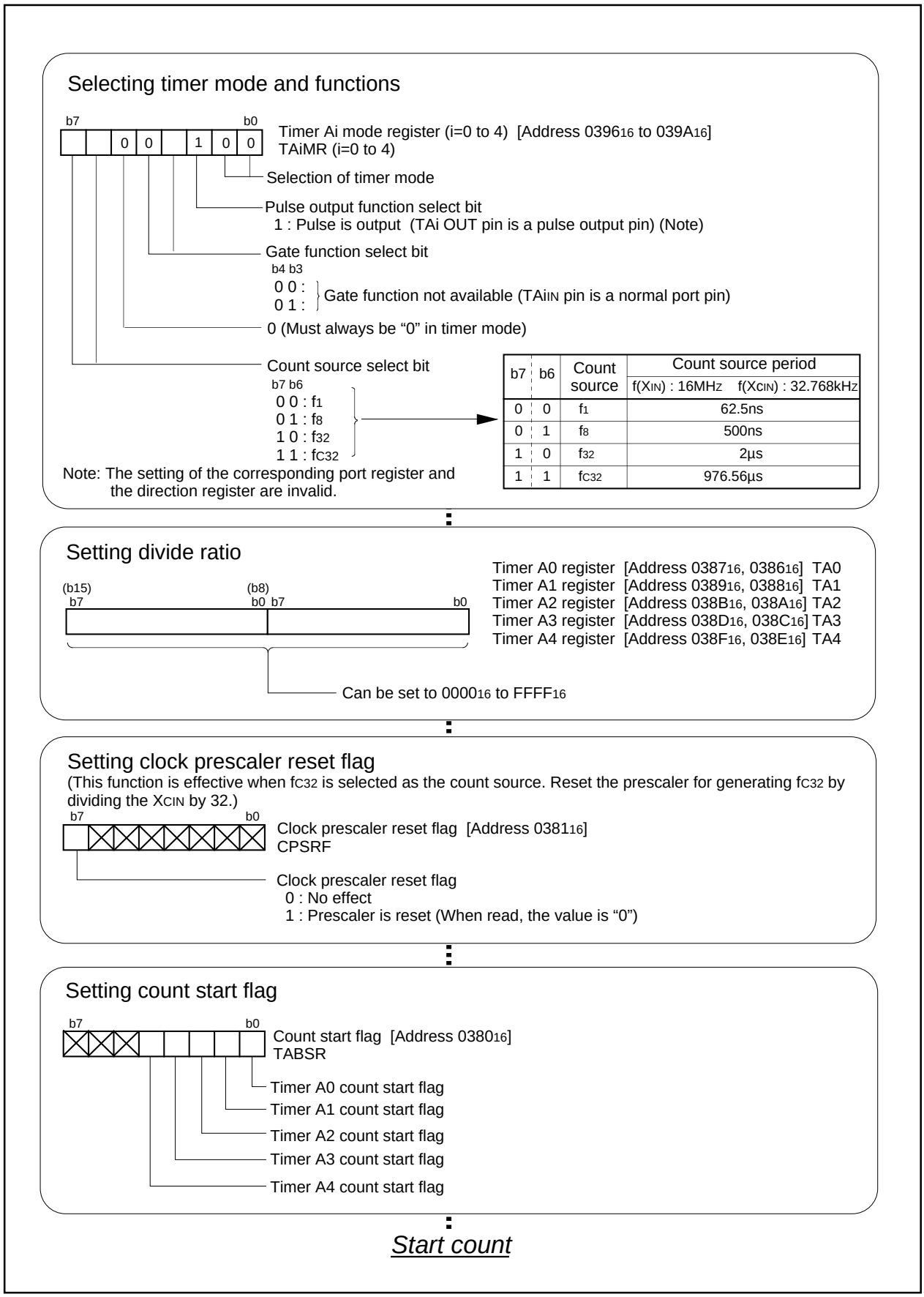

Selecting timer mode and functions
b7 b0
0 0 1 0 0
Timer Ai mode register (i=0 to 4) [Address 039616 to 039A16]
TAiMR (i=0 to 4)
Selection of timer mode
Pulse output function select bit
1 : Pulse is output (TAi OUT pin is a pulse output pin) (Note)
Gate function select bit
b4 b3
0 0 :
0 1 : Gate function not available (TAiIN pin is a normal port pin)
0 (Must always be "0" in timer mode)
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fC32
b7 b6 Count source Count source period
f(XIN) : 16MHz f(XCIN) : 32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fC32 976.56µs
Note: The setting of the corresponding port register and the direction register are invalid.
Setting divide ratio
(b15) (b8)
b7 b0 b7 b0
Timer A0 register [Address 038716, 038616] TA0
Timer A1 register [Address 038916, 038816] TA1
Timer A2 register [Address 038B16, 038A16] TA2
Timer A3 register [Address 038D16, 038C16] TA3
Timer A4 register [Address 038F16, 038E16] TA4
Can be set to 000016 to FFFF16
Setting clock prescaler reset flag
(This function is effective when fC32 is selected as the count source. Reset the prescaler for generating fC32 by dividing the XCIN by 32.)
b7 b0
Clock prescaler reset flag [Address 038116]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")
Setting count start flag
b7 b0
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
Timer A1 count start flag
Timer A2 count start flag
Timer A3 count start flag
Timer A4 count start flag
Start count

Figure 2.2.11. Set-up procedure of timer mode, pulse output function selected

2.2.5 Operation of Timer A (event counter mode, reload type selected)

In event counter mode, choose functions from those listed in Table 2.2.4. Operations of the circled items are described below. Figure 2.2.12 shows the operation timing, and Figure 2.2.13 shows the set-up procedure.

Table 2.2.4. Choosed functions

Item	Set-up		Item	Set-up	
Count source	O	Input signal to TAiIN (counting falling edges)	Pulse output function	O	No pulses output
					Pulses output
		Input signal to TAiIN (counting rising edges)	Count operation type	O	Reload type
					Free-run type
		TAj overflow	Factor for switching between up and down	O	Content of up/down flag
					Input signal to TAiOUT

Note: j = i – 1, but j = 4 when i = 0.

Operation (1) Setting the count start flag to “1” causes the counter to count the falling edges of the count source.

(2) If an underflow occurs, the content of the reload register is reloaded, and the count continues. At this time, the timer Ai interrupt request bit goes to “1”.

(3) If switching from an up count to a down count or vice versa while a count is in progress, the switch takes effect from the next effective edge of the count source.

(4) Setting the count start flag to “0” causes the counter to hold its value and to stop.

(5) If an overflow occurs, the content of the reload register is reloaded, and the count continues. At this time, the timer Ai interrupt request bit goes to “1”.

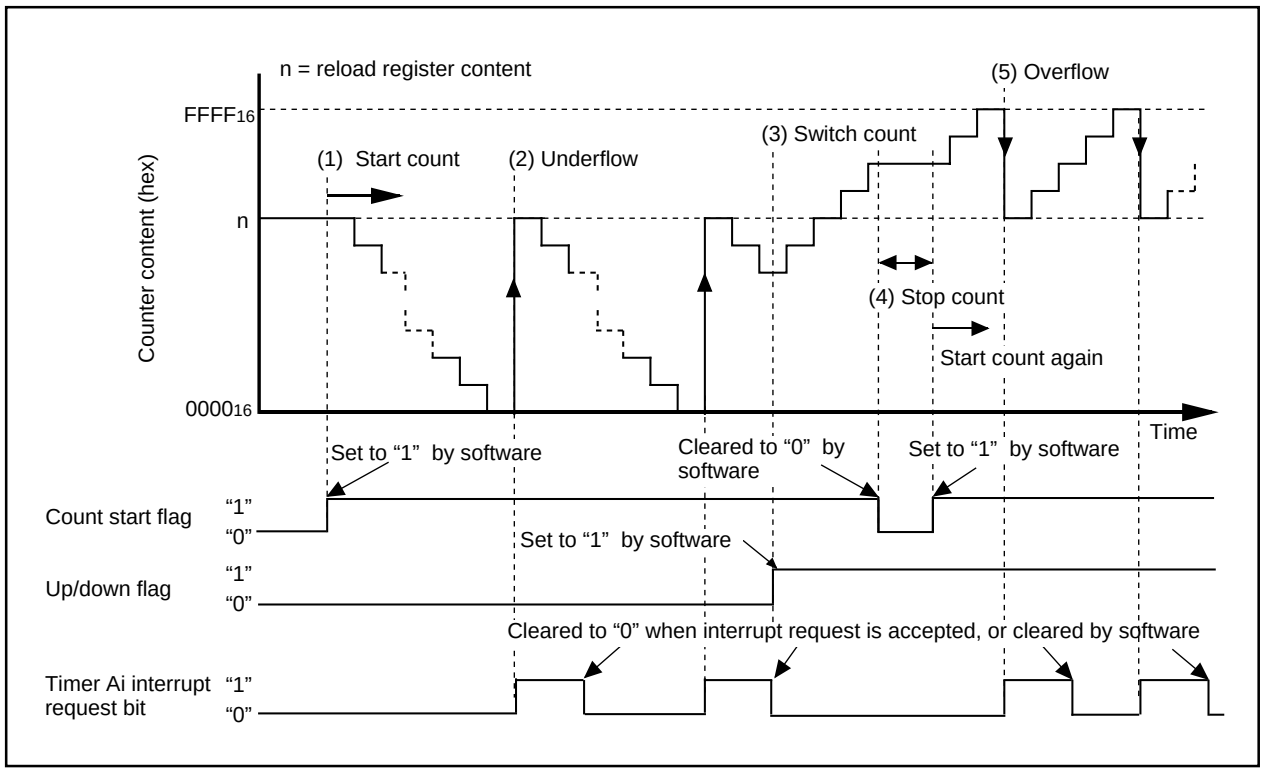


Figure 2.2.12. Operation timing of event counter mode, reload type selected

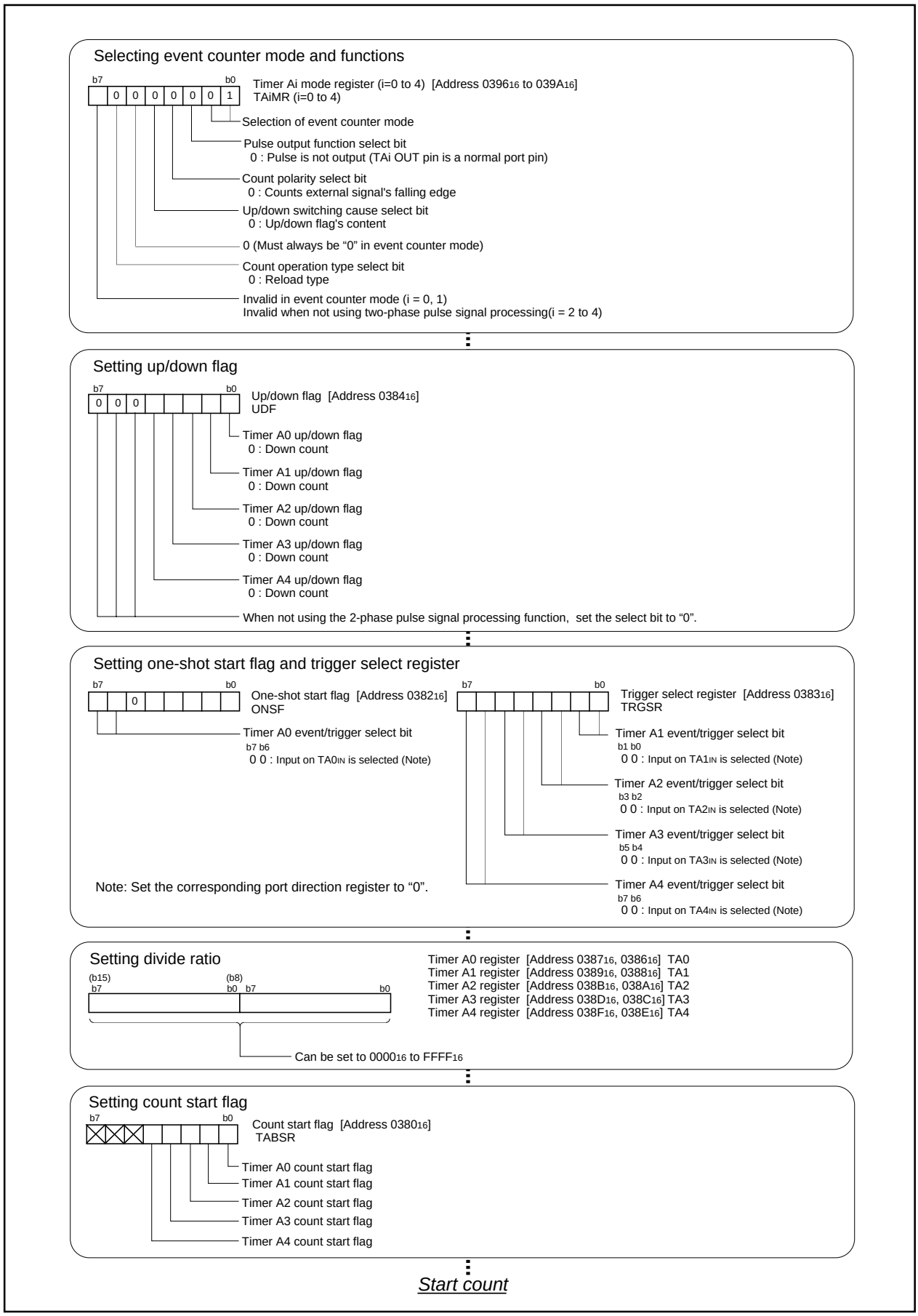

Selecting event counter mode and functions
b7 b0
0 0 0 0 0 0 1
Timer Ai mode register (i=0 to 4) [Address 039616 to 039A16]
TAiMR (i=0 to 4)
Selection of event counter mode
Pulse output function select bit
0 : Pulse is not output (TAi OUT pin is a normal port pin)
Count polarity select bit
0 : Counts external signal's falling edge
Up/down switching cause select bit
0 : Up/down flag's content
0 (Must always be "0" in event counter mode)
Count operation type select bit
0 : Reload type
Invalid in event counter mode (i = 0, 1)
Invalid when not using two-phase pulse signal processing(i = 2 to 4)
Setting up/down flag
b7 b0
0 0 0
Up/down flag [Address 038416]
UDF
Timer A0 up/down flag
0 : Down count
Timer A1 up/down flag
0 : Down count
Timer A2 up/down flag
0 : Down count
Timer A3 up/down flag
0 : Down count
Timer A4 up/down flag
0 : Down count
When not using the 2-phase pulse signal processing function, set the select bit to "0".
Setting one-shot start flag and trigger select register
b7 b0
0
One-shot start flag [Address 038216]
ONSF
Timer A0 event/trigger select bit
b7 b6
0 0 : Input on TA0IN is selected (Note)
b7 b0
Trigger select register [Address 038316]
TRGSR
Timer A1 event/trigger select bit
b1 b0
0 0 : Input on TA1IN is selected (Note)
Timer A2 event/trigger select bit
b3 b2
0 0 : Input on TA2IN is selected (Note)
Timer A3 event/trigger select bit
b5 b4
0 0 : Input on TA3IN is selected (Note)
Timer A4 event/trigger select bit
b7 b6
0 0 : Input on TA4IN is selected (Note)
Note: Set the corresponding port direction register to "0".
Setting divide ratio
(b15) (b8)
b7 b0 b7 b0
Timer A0 register [Address 038716, 038616] TA0
Timer A1 register [Address 038916, 038816] TA1
Timer A2 register [Address 038B16, 038A16] TA2
Timer A3 register [Address 038D16, 038C16] TA3
Timer A4 register [Address 038F16, 038E16] TA4
Can be set to 000016 to FFFF16
Setting count start flag
b7 b0
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
Timer A1 count start flag
Timer A2 count start flag
Timer A3 count start flag
Timer A4 count start flag
Start count

Figure 2.2.13. Set-up procedure of event counter mode, reload type selected

2.2.6 Operation of Timer A (event counter mode, free run type selected)

In event counter mode, choose functions from those listed in Table 2.2.5. Operations of the circled items are described below. Figure 2.2.14 shows the operation timing, and Figure 2.2.15 shows the set-up procedure.

Table 2.2.5. Choosed functions

Item	Set-up		Item	Set-up	
Count source	O	Input signal to TAiIN (counting falling edges)	Pulse output function	O	No pulses output
					Pulses output
		Input signal to TAiIN (counting rising edges)	Count operation type		Reload type
				O	Free-run type
		TAj overflow	Factor for switching between up and down	O	Content of up/down flag
					Input signal to TAiOUT

Note: j = i – 1, but j = 4 when i = 0

Operation

(1) Setting the count start flag to “1” causes the counter to count the falling edges of the count source.

(2) Even if an underflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer Ai interrupt request bit goes to “1”.

(3) If switching from an up count to a down count or vice versa while a count is in progress, the switch takes effect from the next effective edge of the count source.

(4) Even if an overflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer Ai interrupt request bit goes to “1”.

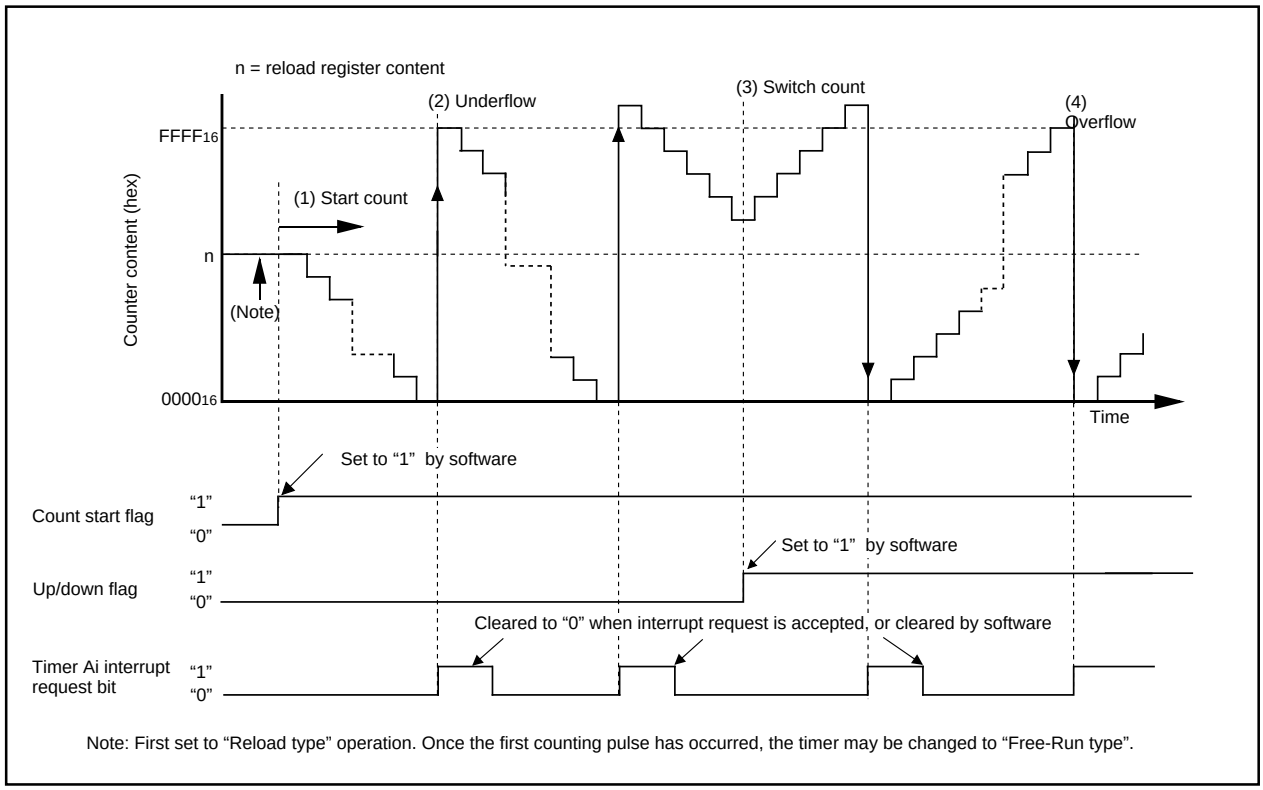


Figure 2.2.14. Operation timing of event counter mode, free run type selected

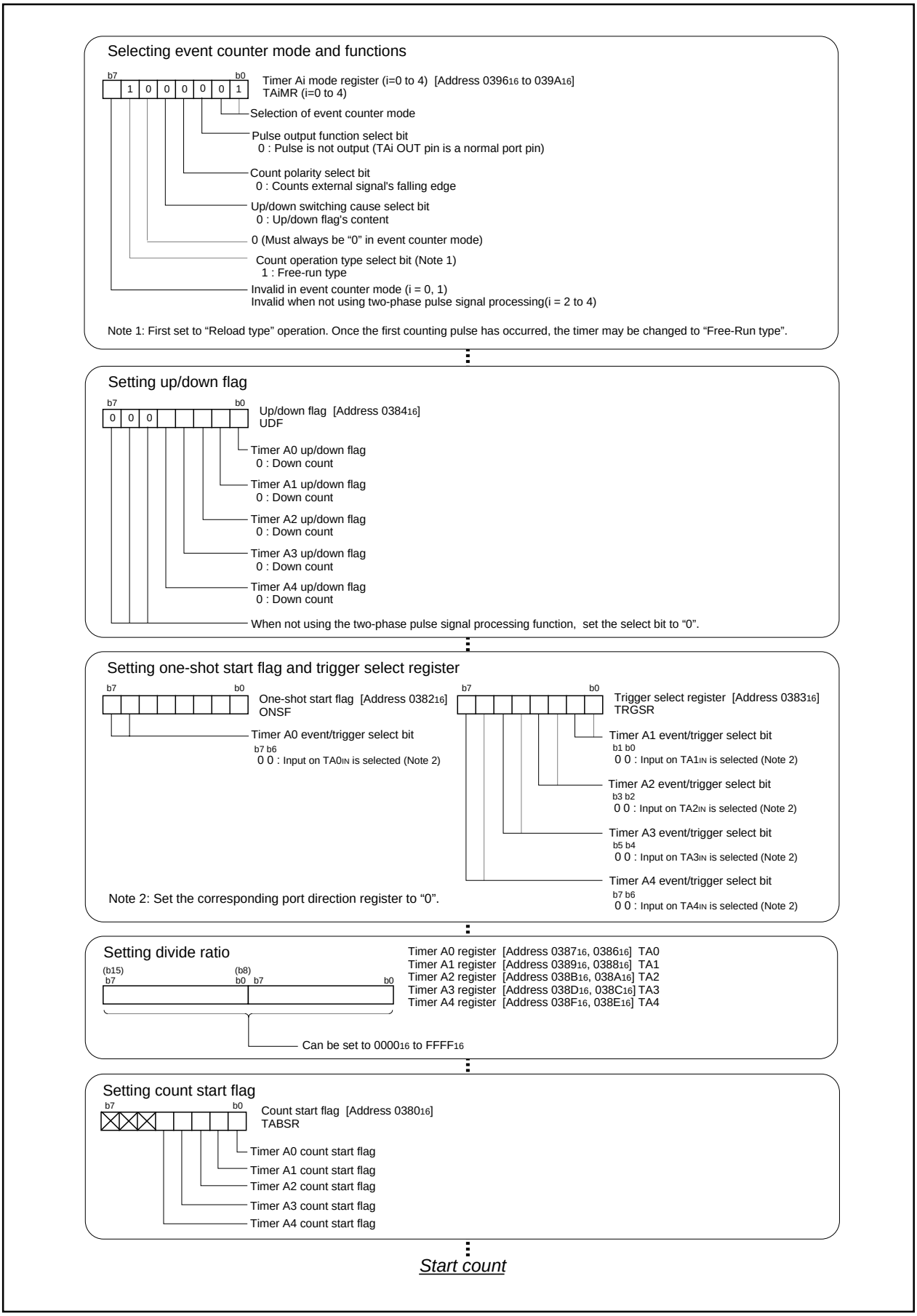

Selecting event counter mode and functions
b7 b0
1 0 0 0 0 0 0 1
Timer Ai mode register (i=0 to 4) [Address 039616 to 039A16]
TAiMR (i=0 to 4)
Selection of event counter mode
Pulse output function select bit
0 : Pulse is not output (TAi OUT pin is a normal port pin)
Count polarity select bit
0 : Counts external signal's falling edge
Up/down switching cause select bit
0 : Up/down flag's content
0 (Must always be "0" in event counter mode)
Count operation type select bit (Note 1)
1 : Free-run type
Invalid in event counter mode (i = 0, 1)
Invalid when not using two-phase pulse signal processing(i = 2 to 4)
Note 1: First set to "Reload type" operation. Once the first counting pulse has occurred, the timer may be changed to "Free-Run type".
Setting up/down flag
b7 b0
0 0 0
Up/down flag [Address 038416]
UDF
Timer A0 up/down flag
0 : Down count
Timer A1 up/down flag
0 : Down count
Timer A2 up/down flag
0 : Down count
Timer A3 up/down flag
0 : Down count
Timer A4 up/down flag
0 : Down count
When not using the two-phase pulse signal processing function, set the select bit to "0".
Setting one-shot start flag and trigger select register
b7 b0
One-shot start flag [Address 038216]
ONSF
Timer A0 event/trigger select bit
b7 b6
0 0 : Input on TA0IN is selected (Note 2)
b7 b0
Trigger select register [Address 038316]
TRGSR
Timer A1 event/trigger select bit
b1 b0
0 0 : Input on TA1IN is selected (Note 2)
Timer A2 event/trigger select bit
b3 b2
0 0 : Input on TA2IN is selected (Note 2)
Timer A3 event/trigger select bit
b5 b4
0 0 : Input on TA3IN is selected (Note 2)
Timer A4 event/trigger select bit
b7 b6
0 0 : Input on TA4IN is selected (Note 2)
Note 2: Set the corresponding port direction register to "0".
Setting divide ratio
(b15) (b8)
b7 b0 b7 b0
Timer A0 register [Address 038716, 038616] TA0
Timer A1 register [Address 038916, 038816] TA1
Timer A2 register [Address 038B16, 038A16] TA2
Timer A3 register [Address 038D16, 038C16] TA3
Timer A4 register [Address 038F16, 038E16] TA4
Can be set to 000016 to FFFF16
Setting count start flag
b7 b0
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
Timer A1 count start flag
Timer A2 count start flag
Timer A3 count start flag
Timer A4 count start flag
Start count

Figure 2.2.15. Set-up procedure of event counter mode, free run type selected

2.2.7 Operation of timer A (two-phase pulse signal process in event counter mode, normal mode selected)

In processing two-phase pulse signals in event counter mode, choose functions from those listed in Table 2.2.6. Operations of the circled items are described below. Figure 2.2.16 shows the operation timing, and Figure 2.2.17 shows the set-up procedure.

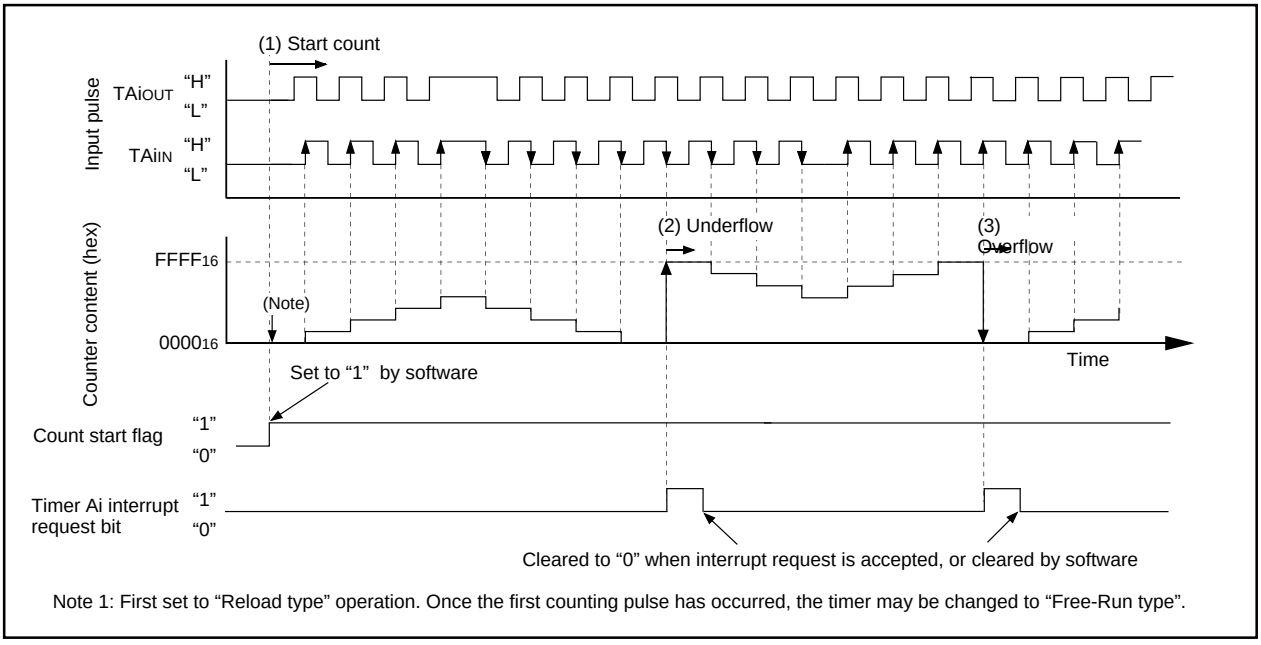
Table 2.2.6. Choosed functions

Item	Set-up	
Count operation type		Reload type
	O	Free run type
Two-phase pulses process (Note)	O	Normal processing
		4-multiplication processing

Note: Timer A3 alone can be selected. Timer A2 is solely used for normal processes, and timer A4 is solely used for 4 multiplication processes.

Operation (1) Setting the count start flag to “1” causes the counter to count effective edges of the count source.
(2) Even if an underflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer Ai interrupt request bit goes to “1”.
(3) Even if an overflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer Ai interrupt request bit goes to “1”.

Note • The up count or down count conditions are as follows:
If a rising edge is present at the TAiIN pin when the input signal level to the TAiOUT pin is “H”, an up count is performed.
If a falling edge is present at the TAiIN pin when the input signal level to the TAiOUT pin is “H”, a down count is performed.
• Set TAiIN pin and TAiOUT pin's port direction register to “0”.



Note 1: First set to “Reload type” operation. Once the first counting pulse has occurred, the timer may be changed to “Free-Run type”.

Figure 2.2.16. Operation timing of two-phase pulse signal process in event counter mode, normal mode selected

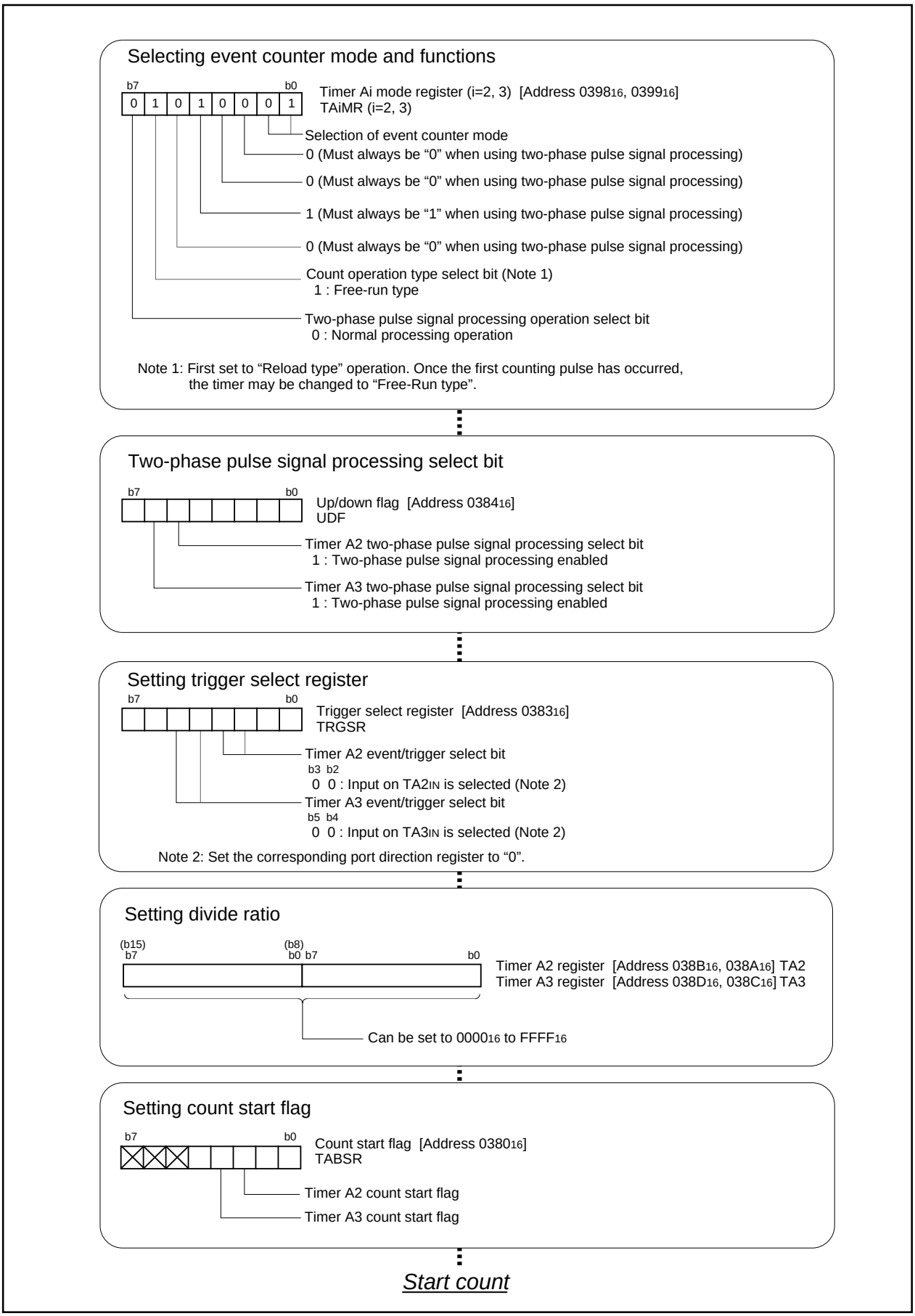

Selecting event counter mode and functions
b7 b0
0 1 0 1 0 0 0 1
Timer Ai mode register (i=2, 3) [Address 039816, 039916]
TAiMR (i=2, 3)
Selection of event counter mode
0 (Must always be “0” when using two-phase pulse signal processing)
0 (Must always be “0” when using two-phase pulse signal processing)
1 (Must always be “1” when using two-phase pulse signal processing)
0 (Must always be “0” when using two-phase pulse signal processing)
Count operation type select bit (Note 1)
1 : Free-run type
Two-phase pulse signal processing operation select bit
0 : Normal processing operation
Note 1: First set to “Reload type” operation. Once the first counting pulse has occurred, the timer may be changed to “Free-Run type”.
Two-phase pulse signal processing select bit
b7 b0
Up/down flag [Address 038416]
UDF
Timer A2 two-phase pulse signal processing select bit
1 : Two-phase pulse signal processing enabled
Timer A3 two-phase pulse signal processing select bit
1 : Two-phase pulse signal processing enabled
Setting trigger select register
b7 b0
Trigger select register [Address 038316]
TRGSR
Timer A2 event/trigger select bit
b3 b2
0 0 : Input on TA2IN is selected (Note 2)
Timer A3 event/trigger select bit
b5 b4
0 0 : Input on TA3IN is selected (Note 2)
Note 2: Set the corresponding port direction register to “0”.
Setting divide ratio
(b15) (b8)
b7 b0 b7 b0
Timer A2 register [Address 038B16, 038A16] TA2
Timer A3 register [Address 038D16, 038C16] TA3
Can be set to 000016 to FFFF16
Setting count start flag
b7 b0
Count start flag [Address 038016]
TABSR
Timer A2 count start flag
Timer A3 count start flag
Start count

Figure 2.2.17. Set-up procedure of two-phase pulse signal process in event counter mode, normal mode selected

2.2.8 Operation of timer A (two-phase pulse signal process in event counter mode, multiply-by-4 mode selected)

In processing two-phase pulse signals in event counter mode, choose functions from those listed in Table 2.2.7. Operations of the circled items are described below. Figure 2.2.18 shows the operation timing, and Figure 2.2.19 shows the set-up procedure.

Table 2.2.7. Choosed functions

Item	Set-up		Item	Set-up	
Count operation type		Reload type	Processing two- phase pulses (Note)		Normal processing
	O	Free run type		O	4-multiplication processing

Note: Timer A3 alone can be selected. Timer A2 is solely used for normal processes, and timer A4 is solely used for 4-multiplication processes.

Operation

(1) Setting the count start flag to “1” causes the counter to count effective edges of the count source.
(2) Even if an underflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the interrupt request bit goes to “1”.
(3) Even if an overflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the interrupt request bit goes to “1”.

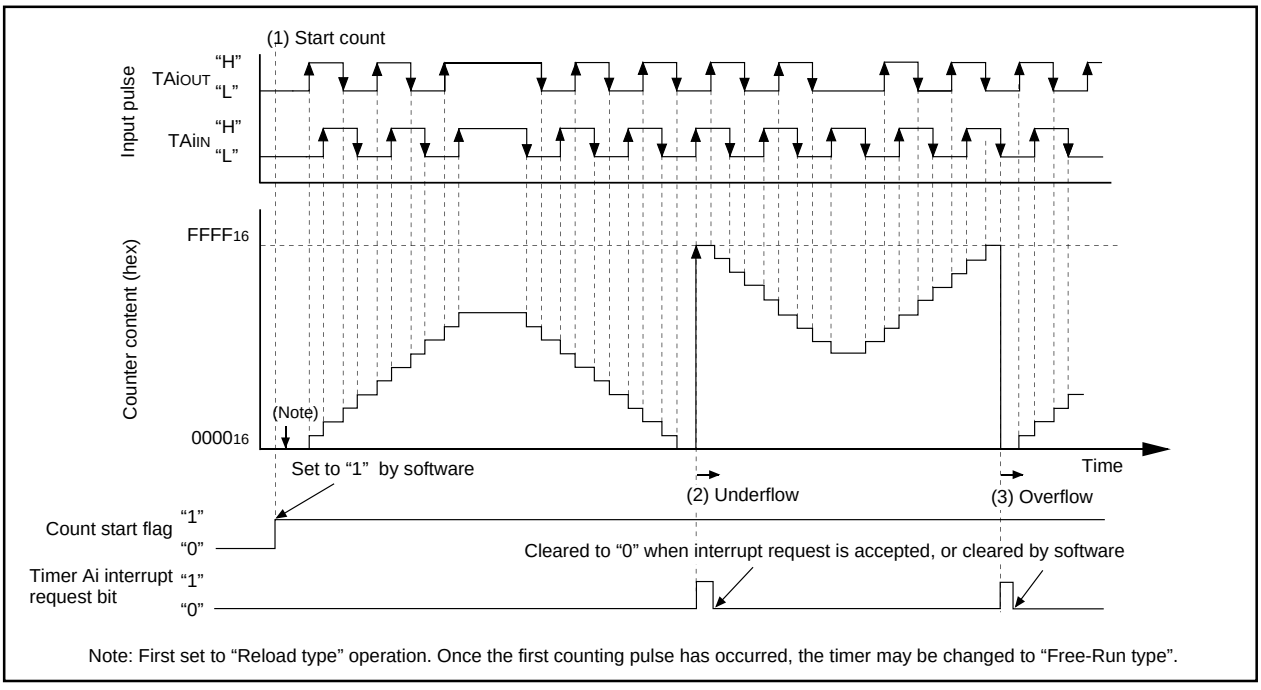
Note

• The up count or down count conditions are as follows:

Table 2.2.8. The up count or down count conditions

	Input signal to the TAiOUT pin	Input signal to the TAiIN pin		Input signal to the TAiOUT pin	Input signal to the TAiIN pin
Up count	“H” level	Rising	Down count	“H” level	Falling
	“L” level	Falling		“L” level	Rising
	Rising	“L” level		Rising	“H” level
	Falling	“H” level		Falling	“L” level

• Set TAiIN pin and TAiOUT pin's port direction register to “0”.



Note: First set to “Reload type” operation. Once the first counting pulse has occurred, the timer may be changed to “Free-Run type”.

Figure 2.2.18. Operation timing of two-phase pulse signal process in event counter mode, multiply-by-4 mode selected

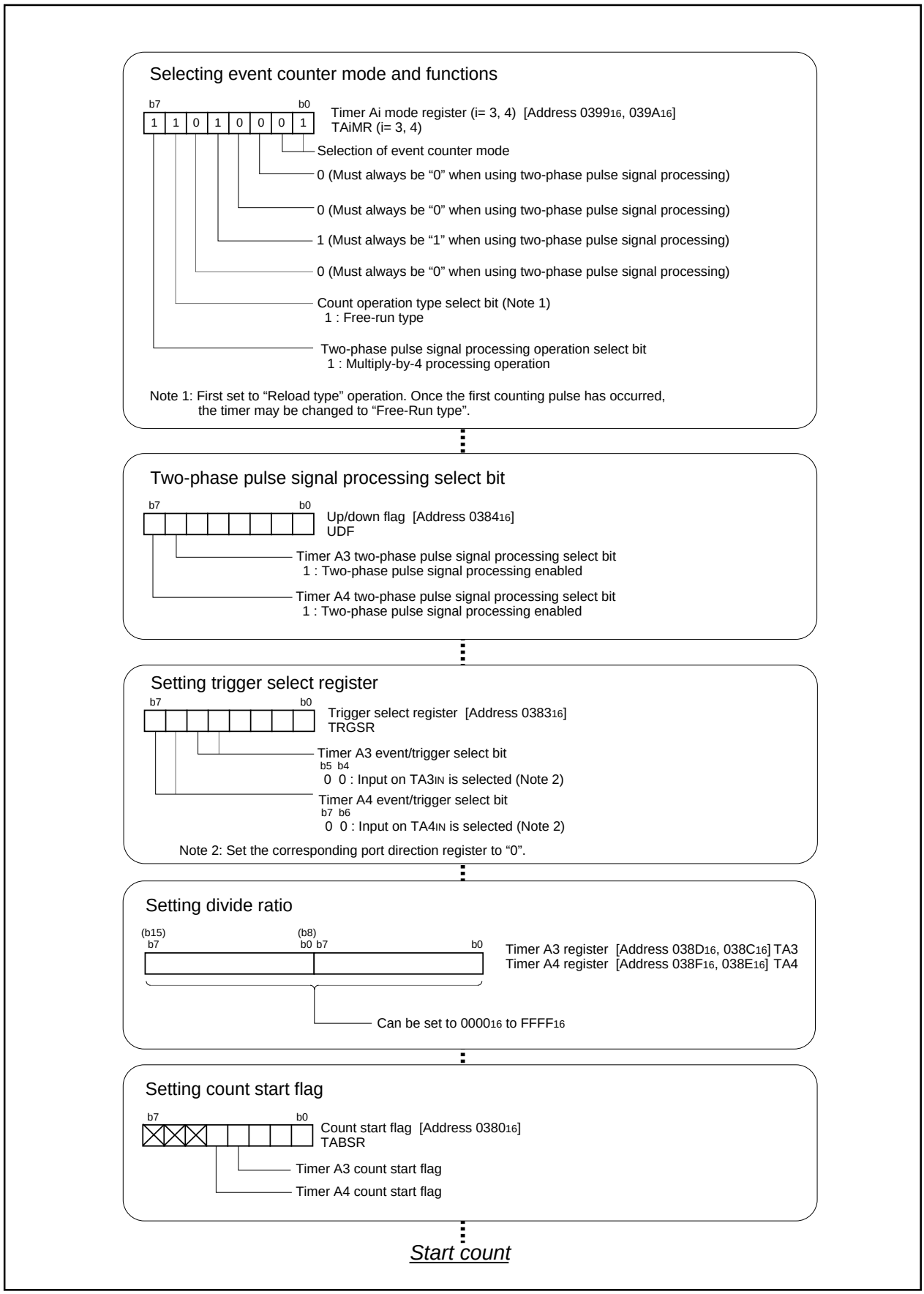

Selecting event counter mode and functions
b7 b0
1 1 0 1 0 0 0 1
Timer Ai mode register (i= 3, 4) [Address 039916, 039A16]
TAiMR (i= 3, 4)
Selection of event counter mode
0 (Must always be "0" when using two-phase pulse signal processing)
0 (Must always be "0" when using two-phase pulse signal processing)
1 (Must always be "1" when using two-phase pulse signal processing)
0 (Must always be "0" when using two-phase pulse signal processing)
Count operation type select bit (Note 1)
1 : Free-run type
Two-phase pulse signal processing operation select bit
1 : Multiply-by-4 processing operation
Note 1: First set to "Reload type" operation. Once the first counting pulse has occurred, the timer may be changed to "Free-Run type".
Two-phase pulse signal processing select bit
b7 b0
Up/down flag [Address 038416]
UDF
Timer A3 two-phase pulse signal processing select bit
1 : Two-phase pulse signal processing enabled
Timer A4 two-phase pulse signal processing select bit
1 : Two-phase pulse signal processing enabled
Setting trigger select register
b7 b0
Trigger select register [Address 038316]
TRGSR
Timer A3 event/trigger select bit
b5 b4
0 0 : Input on TA3IN is selected (Note 2)
Timer A4 event/trigger select bit
b7 b6
0 0 : Input on TA4IN is selected (Note 2)
Note 2: Set the corresponding port direction register to "0".
Setting divide ratio
(b15) (b8)
b7 b0 b7 b0
Timer A3 register [Address 038D16, 038C16] TA3
Timer A4 register [Address 038F16, 038E16] TA4
Can be set to 000016 to FFFF16
Setting count start flag
b7 b0
Count start flag [Address 038016]
TABSR
Timer A3 count start flag
Timer A4 count start flag
Start count

Figure 2.2.19. Set-up procedure of two-phase pulse signal process in event counter mode, multiply-by-4 mode selected

2.2.9 Operation of Timer A (one-shot timer mode)

In one-shot timer mode, choose functions from those listed in Table 2.2.9. Operations of the circled items are described below. Figure 2.2.20 shows the operation timing, and Figures 2.2.21 shows the set-up procedure.

Table 2.2.9. Choosed functions

Item	Set-up	
Count source	**O**	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
Pulse output function		No pulses output
	O	Pulses output
Count start condition		External trigger input (falling edge of input signal to the TAi$_{IN}$ pin)
		External trigger input (rising edge of input signal to the TAi$_{IN}$ pin)
		Timer overflow (TAj/TAk overflow)
	O	Writing “1” to the one-shot start flag

Note: j = i – 1, but j = 4 when i = 0; k = i + 1, but k = 0 when i = 4.

Operation

(1) Setting the one-shot start flag to “1” with the count start flag set to “1” causes the counter to perform a down count on the count source. At this time, the TAi$_{OUT}$ pin outputs an “H” level.

(2) The instant the value of the counter becomes “0000$_{16}$”, the TAi$_{OUT}$ pin outputs an “L” level, and the counter reloads the content of the reload register and stops counting. At this time, the timer Ai interrupt request bit goes to “1”.

(3) If a trigger occurs while a count is in progress, the counter reloads the value in the reload register again and continues counting. The reload timing is in step with the next count source input after the trigger.

(4) Setting the count start flag to “0” causes the counter to stop and to reload the content of the reload register. Also, the TAi$_{OUT}$ pin outputs an “L” level. At this time, the timer Ai interrupt request bit goes to “1”.

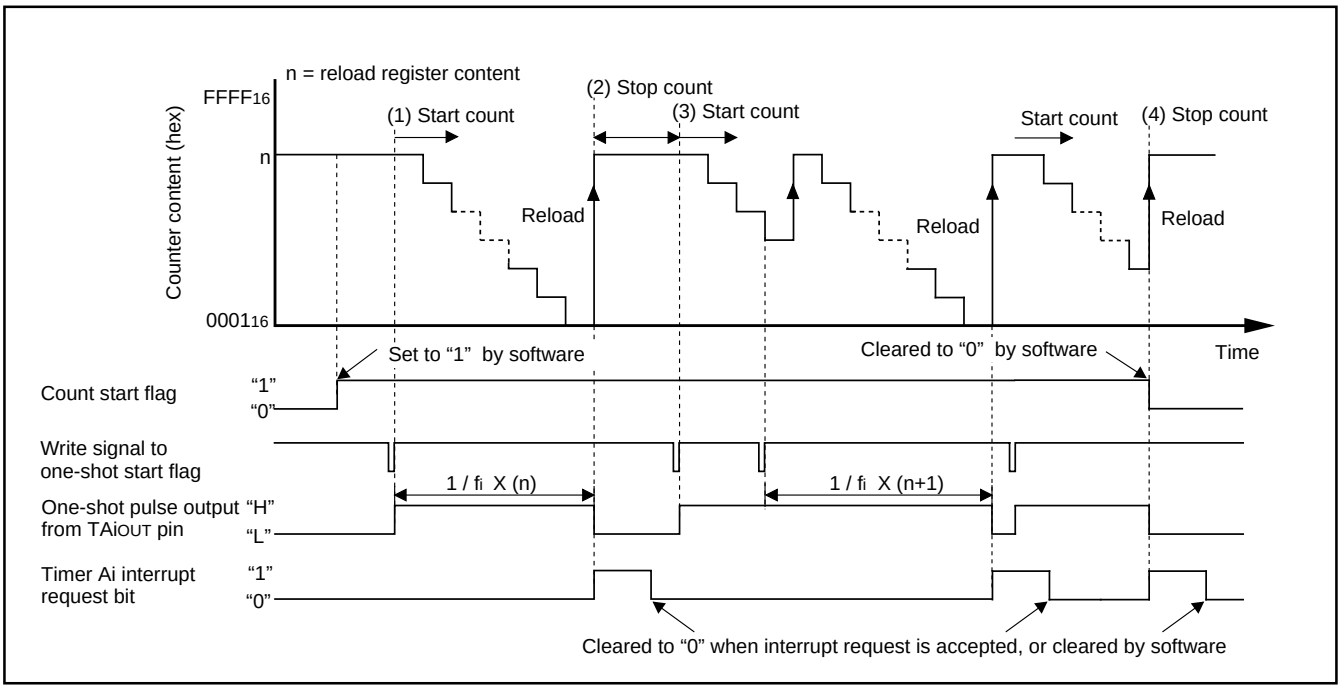


Figure 2.2.20. Operation timing of one-shot mode

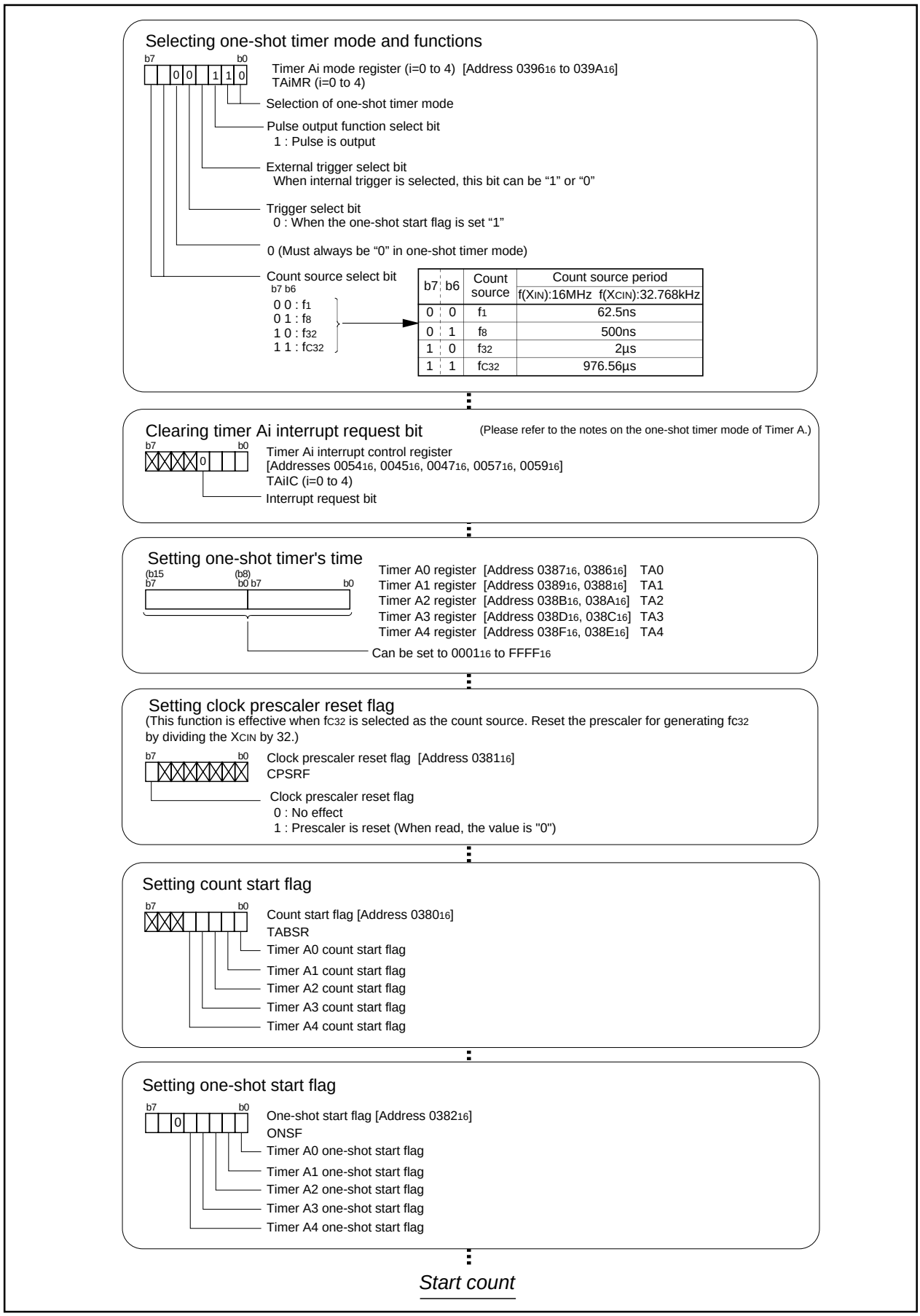

Selecting one-shot timer mode and functions
b7 b0
0 0 1 1 0
Timer Ai mode register (i=0 to 4) [Address 039616 to 039A16]
TAiMR (i=0 to 4)
Selection of one-shot timer mode
Pulse output function select bit
1 : Pulse is output
External trigger select bit
When internal trigger is selected, this bit can be "1" or "0"
Trigger select bit
0 : When the one-shot start flag is set "1"
0 (Must always be "0" in one-shot timer mode)
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fc32
b7 b6 Count source Count source period
f(XIN):16MHz f(XCIN):32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fc32 976.56µs
Clearing timer Ai interrupt request bit
(Please refer to the notes on the one-shot timer mode of Timer A.)
b7 b0
0
Timer Ai interrupt control register
[Addresses 005416, 004516, 004716, 005716, 005916]
TAiIC (i=0 to 4)
Interrupt request bit
Setting one-shot timer's time
(b15) (b8)
b7 b0 b7 b0
Timer A0 register [Address 038716, 038616] TA0
Timer A1 register [Address 038916, 038816] TA1
Timer A2 register [Address 038B16, 038A16] TA2
Timer A3 register [Address 038D16, 038C16] TA3
Timer A4 register [Address 038F16, 038E16] TA4
Can be set to 000116 to FFFF16
Setting clock prescaler reset flag
(This function is effective when fc32 is selected as the count source. Reset the prescaler for generating fc32 by dividing the XCIN by 32.)
b7 b0
Clock prescaler reset flag [Address 038116]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")
Setting count start flag
b7 b0
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
Timer A1 count start flag
Timer A2 count start flag
Timer A3 count start flag
Timer A4 count start flag
Setting one-shot start flag
b7 b0
0
One-shot start flag [Address 038216]
ONSF
Timer A0 one-shot start flag
Timer A1 one-shot start flag
Timer A2 one-shot start flag
Timer A3 one-shot start flag
Timer A4 one-shot start flag
Start count

Figure 2.2.21. Set-up procedure of one-shot mode

2.2.10 Operation of Timer A (pulse width modulation mode, 16-bit PWM mode selected)

In pulse width modulation mode, choose functions from those listed in Table 2.2.10. Operations of the circled items are described below. Figure 2.2.22 shows the operation timing, and Figures 2.2.23 and 2.2.24 show the set-up procedure.

Table 2.2.10. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
PWM mode	O	16-bit PWM
		8-bit PWM
Count start condition		External trigger input (falling edge of input signal to the TAiIN pin)
	O	External trigger input (rising edge of input signal to the TAiIN pin)
		Timer overflow (TAj/TAk overflow)

Note: j = i – 1, but j = 4 when i = 0; k = i + 1, but k = 0 when i = 4.

Operation

(1) If the TAiIN pin input level changes from “L” to “H” with the count start flag set to “1”, the counter performs a down count on the count source. Also, the TAiOUT pin outputs an “H” level.

(2) The TAiOUT pin output level changes from “H” to “L” when a set time period elapses. At this time, the timer Ai interrupt request bit goes to “1”.

(3) The counter reloads the content of the reload register every time PWM pulses are output for one cycle, and continues counting.

(4) Setting the count start flag to “0” causes the counter to hold its value and to stop. Also, the TAiOUT outputs an “L” level.

Note

- The period of PWM pulses becomes $(2^{16} – 1)/fi$, and the “H” level pulse width becomes n/fi. If the timer Ai register is set to “0000_{16}”, the pulse width modulator does not work, and the the TAiOUT pin output level remains at “L”.
  (fi : frequency of the count source f_1, f_8, f_{32}, fc_{32}; n : value of the timer)
- Set TAiIN pin's port direction register to “0”.

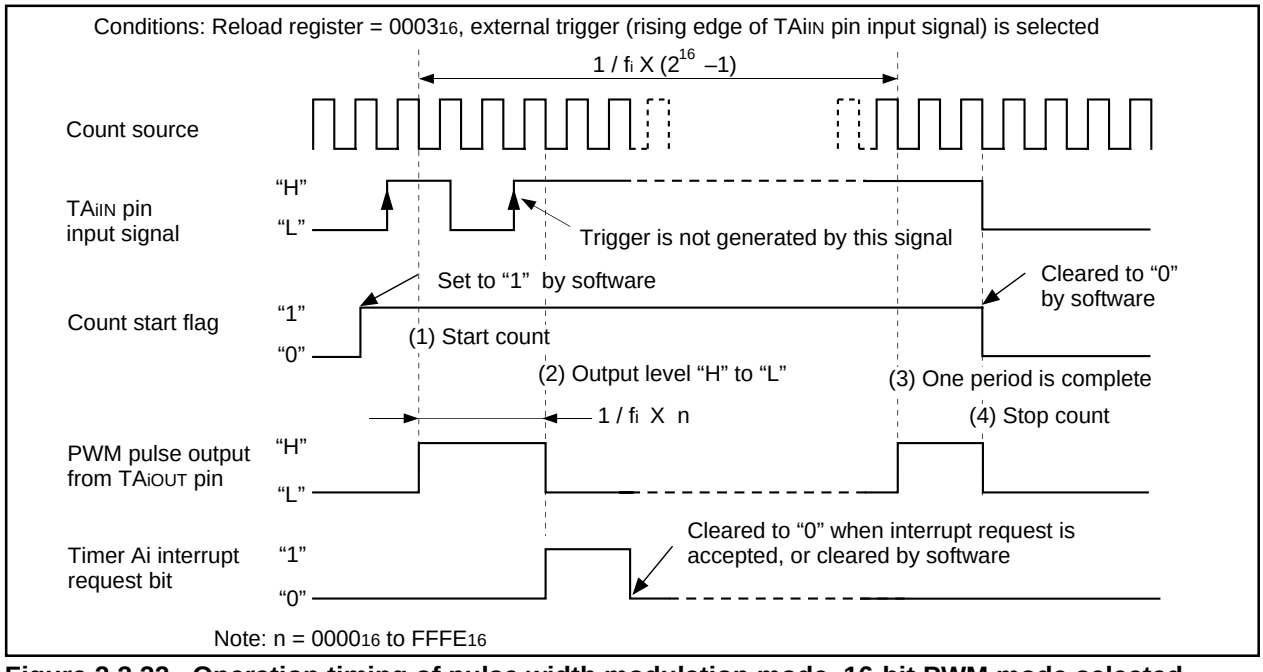

Conditions: Reload register = 0003₁₆, external trigger (rising edge of TAiIN pin input signal) is selected
1 / fi X (2^16 –1)
Count source
TAiIN pin input signal
“H”
“L”
Trigger is not generated by this signal
Set to “1” by software
Cleared to “0” by software
Count start flag
“1”
“0”
(1) Start count
(2) Output level “H” to “L”
(3) One period is complete
1 / fi X n
(4) Stop count
PWM pulse output from TAiOUT pin
“H”
“L”
Timer Ai interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software
Note: n = 0000₁₆ to FFFE₁₆

Figure 2.2.22. Operation timing of pulse width modulation mode, 16-bit PWM mode selected

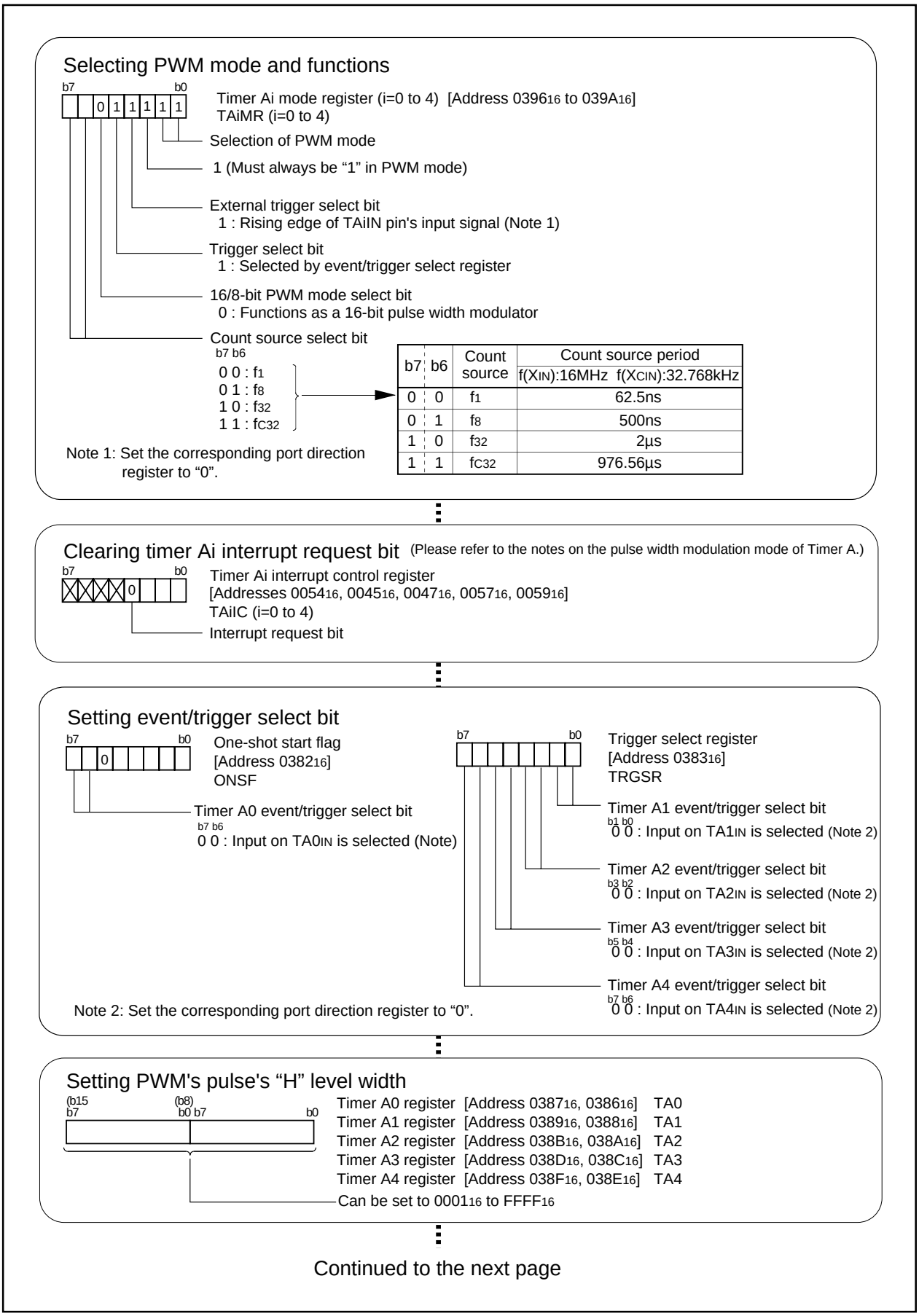

Selecting PWM mode and functions
b7 b0
0 1 1 1 1 1
Timer Ai mode register (i=0 to 4) [Address 039616 to 039A16]
TAiMR (i=0 to 4)
Selection of PWM mode
1 (Must always be "1" in PWM mode)
External trigger select bit
1 : Rising edge of TAiIN pin's input signal (Note 1)
Trigger select bit
1 : Selected by event/trigger select register
16/8-bit PWM mode select bit
0 : Functions as a 16-bit pulse width modulator
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fC32
b7 b6 Count source Count source period
f(XIN):16MHz f(XCIN):32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fC32 976.56µs
Note 1: Set the corresponding port direction register to "0".
Clearing timer Ai interrupt request bit (Please refer to the notes on the pulse width modulation mode of Timer A.)
b7 b0
0
Timer Ai interrupt control register
[Addresses 005416, 004516, 004716, 005716, 005916]
TAiIC (i=0 to 4)
Interrupt request bit
Setting event/trigger select bit
b7 b0
0
One-shot start flag
[Address 038216]
ONSF
Timer A0 event/trigger select bit
b7 b6
0 0 : Input on TA0IN is selected (Note)
b7 b0
Trigger select register
[Address 038316]
TRGSR
Timer A1 event/trigger select bit
b1 b0
0 0 : Input on TA1IN is selected (Note 2)
Timer A2 event/trigger select bit
b3 b2
0 0 : Input on TA2IN is selected (Note 2)
Timer A3 event/trigger select bit
b5 b4
0 0 : Input on TA3IN is selected (Note 2)
Timer A4 event/trigger select bit
b7 b6
0 0 : Input on TA4IN is selected (Note 2)
Note 2: Set the corresponding port direction register to "0".
Setting PWM's pulse's "H" level width
(b15) (b8)
b7 b0 b7 b0
Timer A0 register [Address 038716, 038616] TA0
Timer A1 register [Address 038916, 038816] TA1
Timer A2 register [Address 038B16, 038A16] TA2
Timer A3 register [Address 038D16, 038C16] TA3
Timer A4 register [Address 038F16, 038E16] TA4
Can be set to 000116 to FFFF16
Continued to the next page

Figure 2.2.23. Set-up procedure of pulse width modulation mode, 16-bit PWM mode selected (1)

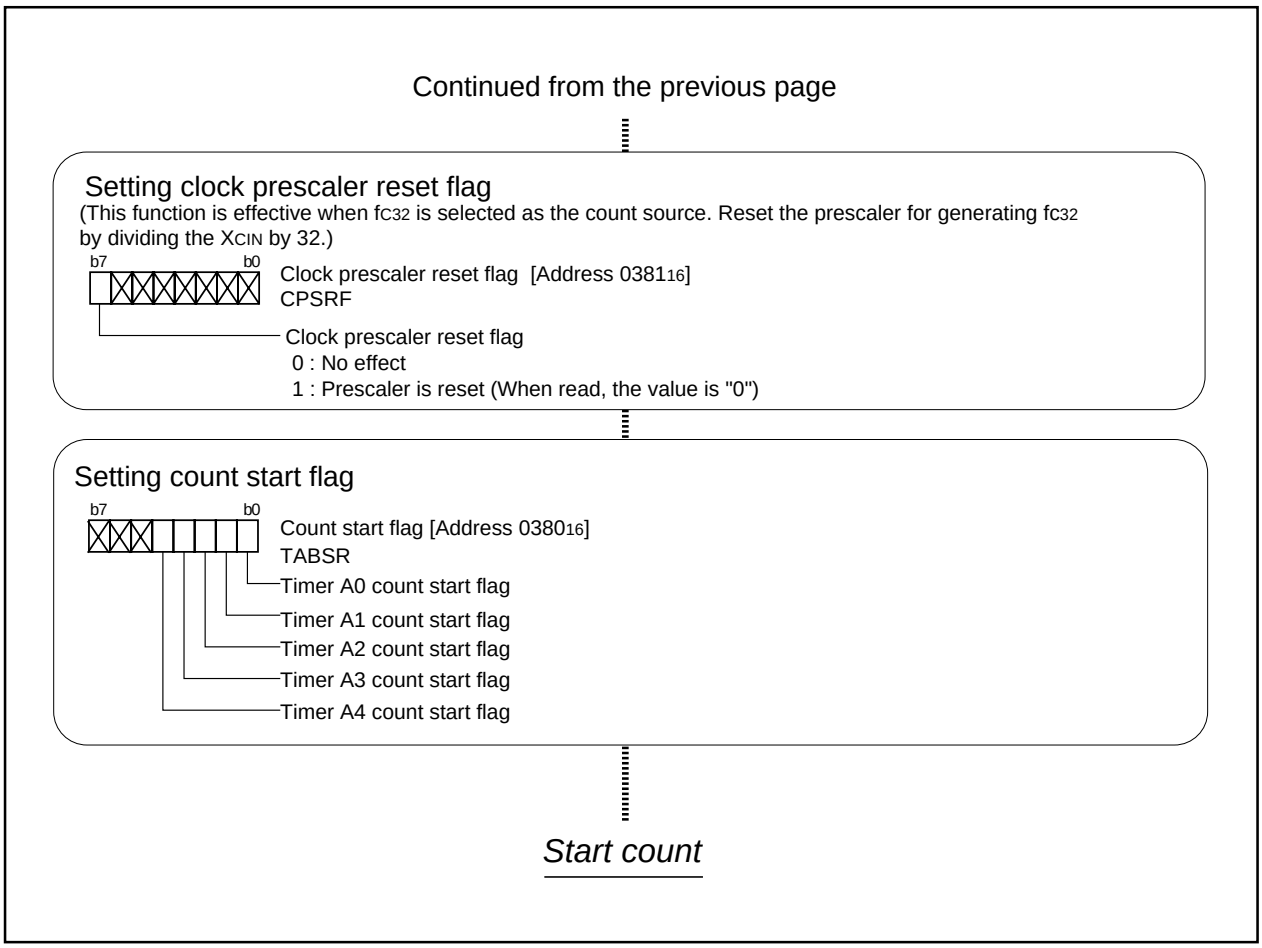

Continued from the previous page
Setting clock prescaler reset flag
(This function is effective when fC32 is selected as the count source. Reset the prescaler for generating fC32 by dividing the XCIN by 32.)
b7
b0
Clock prescaler reset flag [Address 038116]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")
Setting count start flag
b7
b0
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
Timer A1 count start flag
Timer A2 count start flag
Timer A3 count start flag
Timer A4 count start flag
Start count

Figure 2.2.24. Set-up procedure of pulse width modulation mode, 16-bit PWM mode selected (2)

2.2.11 Operation of Timer A (pulse width modulation mode, 8-bit PWM mode selected)

In pulse width modulation mode, choose functions from those listed in Table 2.2.11. Operations of the circled items are described below. Figure 2.2.25 shows the operation timing, and Figures 2.2.26 and 2.2.27 show the set-up procedure.

Table 2.2.11. Choosed functions

Item	Set-up	
Count source	**O**	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
PWM mode		16-bit PWM
	O	8-bit PWM
Count start condition	**O**	External trigger input (falling edge of input signal to the TAiIN pin)
		External trigger input (rising edge of input signal to the TAiIN pin)
		Timer overflow (TAj/TAk overflow)

Note: j = i – 1, but j = 4 when i = 0; k = i + 1, but k = 0 when i = 4.

Operation (1) If the TAiIN pin input level changes from "H" to "L" with the count start flag set to "1", the counter performs a down count on the count source. Also, the TAiOUT pin outputs an "H" level.

(2) The TAiOUT pin output level changes from "H" to "L" when a set time period elapses. At this time, the timer Ai interrupt request bit goes to "1".

(3) The counter reloads the content of the reload register every time PWM pulses are output for one cycle, and continues counting.

(4) Setting the count start flag to "0" causes the counter to hold its value and to stop. Also, the TAiOUT pin outputs an "L" level.

Note
- The period of PWM pulses becomes (m + 1) X (2^8 – 1) / fi, and the "H" level pulse width becomes n X (m + 1) / fi. If "00$_{16}$" is set in the eight higher-order bits of the timer Ai register, the pulse width modulator does not work, and the the TAiOUT pin output level remains at "L".
  (fi : frequency of the count source f_1, f_8, f_{32}, fc_{32}; n : value of the timer)
- When a trigger is generated, the TAiout pin outputs "L" level of same amplitude as "H" level of the set PWM pulse, after which it starts PWM pulse output.
- Set TAiIN pin's port direction register to "0".

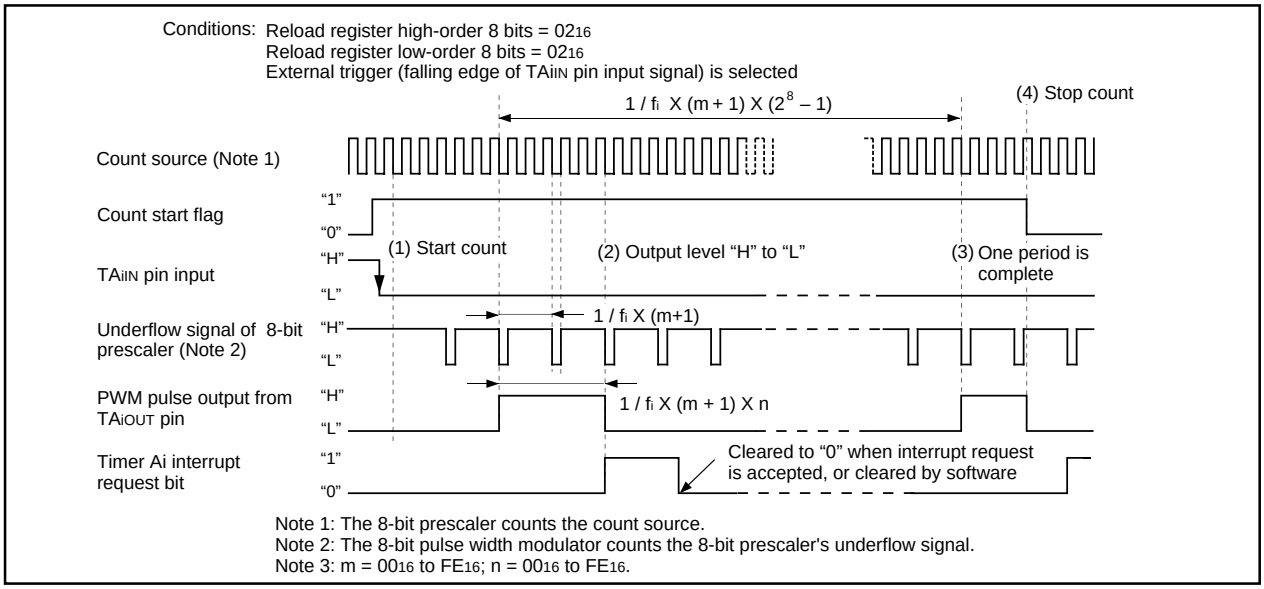

Conditions:
Reload register high-order 8 bits = 02₁₆
Reload register low-order 8 bits = 02₁₆
External trigger (falling edge of TAiIN pin input signal) is selected
1 / fi X (m + 1) X (2^8 – 1)
(4) Stop count
Count source (Note 1)
Count start flag
"1"
"0"
TAiIN pin input
"H"
"L"
(1) Start count
(2) Output level "H" to "L"
(3) One period is complete
Underflow signal of 8-bit prescaler (Note 2)
1 / fi X (m+1)
PWM pulse output from TAiOUT pin
1 / fi X (m + 1) X n
Timer Ai interrupt request bit
Cleared to "0" when interrupt request is accepted, or cleared by software
Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00₁₆ to FE₁₆; n = 00₁₆ to FE₁₆.

Figure 2.2.25. Operation timing of pulse width modulation mode, with 8-bit PWM mode selected

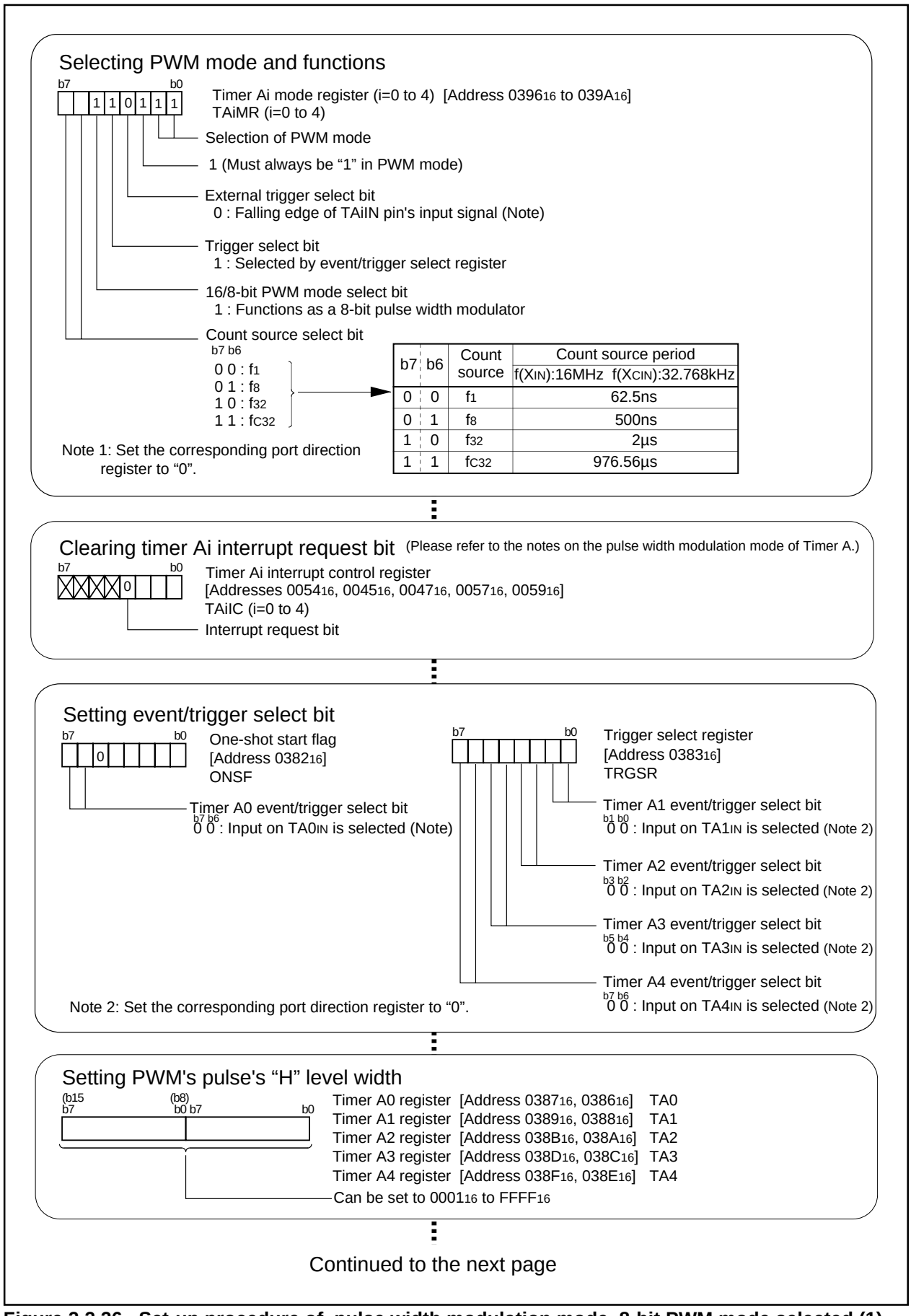

Selecting PWM mode and functions
b7 b0
1 1 0 1 1 1
Timer Ai mode register (i=0 to 4) [Address 039616 to 039A16]
TAiMR (i=0 to 4)
Selection of PWM mode
1 (Must always be "1" in PWM mode)
External trigger select bit
0 : Falling edge of TAiIN pin's input signal (Note)
Trigger select bit
1 : Selected by event/trigger select register
16/8-bit PWM mode select bit
1 : Functions as a 8-bit pulse width modulator
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fC32
b7 b6 Count source Count source period f(XIN):16MHz f(XCIN):32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fC32 976.56µs
Note 1: Set the corresponding port direction register to "0".
Clearing timer Ai interrupt request bit (Please refer to the notes on the pulse width modulation mode of Timer A.)
b7 b0
0
Timer Ai interrupt control register
[Addresses 005416, 004516, 004716, 005716, 005916]
TAiIC (i=0 to 4)
Interrupt request bit
Setting event/trigger select bit
b7 b0
0
One-shot start flag
[Address 038216]
ONSF
Timer A0 event/trigger select bit
b7 b6
0 0 : Input on TA0IN is selected (Note)
b7 b0
Trigger select register
[Address 038316]
TRGSR
Timer A1 event/trigger select bit
b1 b0
0 0 : Input on TA1IN is selected (Note 2)
Timer A2 event/trigger select bit
b3 b2
0 0 : Input on TA2IN is selected (Note 2)
Timer A3 event/trigger select bit
b5 b4
0 0 : Input on TA3IN is selected (Note 2)
Timer A4 event/trigger select bit
b7 b6
0 0 : Input on TA4IN is selected (Note 2)
Note 2: Set the corresponding port direction register to "0".
Setting PWM's pulse's "H" level width
(b15) (b8)
b7 b0 b7 b0
Timer A0 register [Address 038716, 038616] TA0
Timer A1 register [Address 038916, 038816] TA1
Timer A2 register [Address 038B16, 038A16] TA2
Timer A3 register [Address 038D16, 038C16] TA3
Timer A4 register [Address 038F16, 038E16] TA4
Can be set to 000116 to FFFF16
Continued to the next page

Figure 2.2.26. Set-up procedure of pulse width modulation mode, 8-bit PWM mode selected (1)

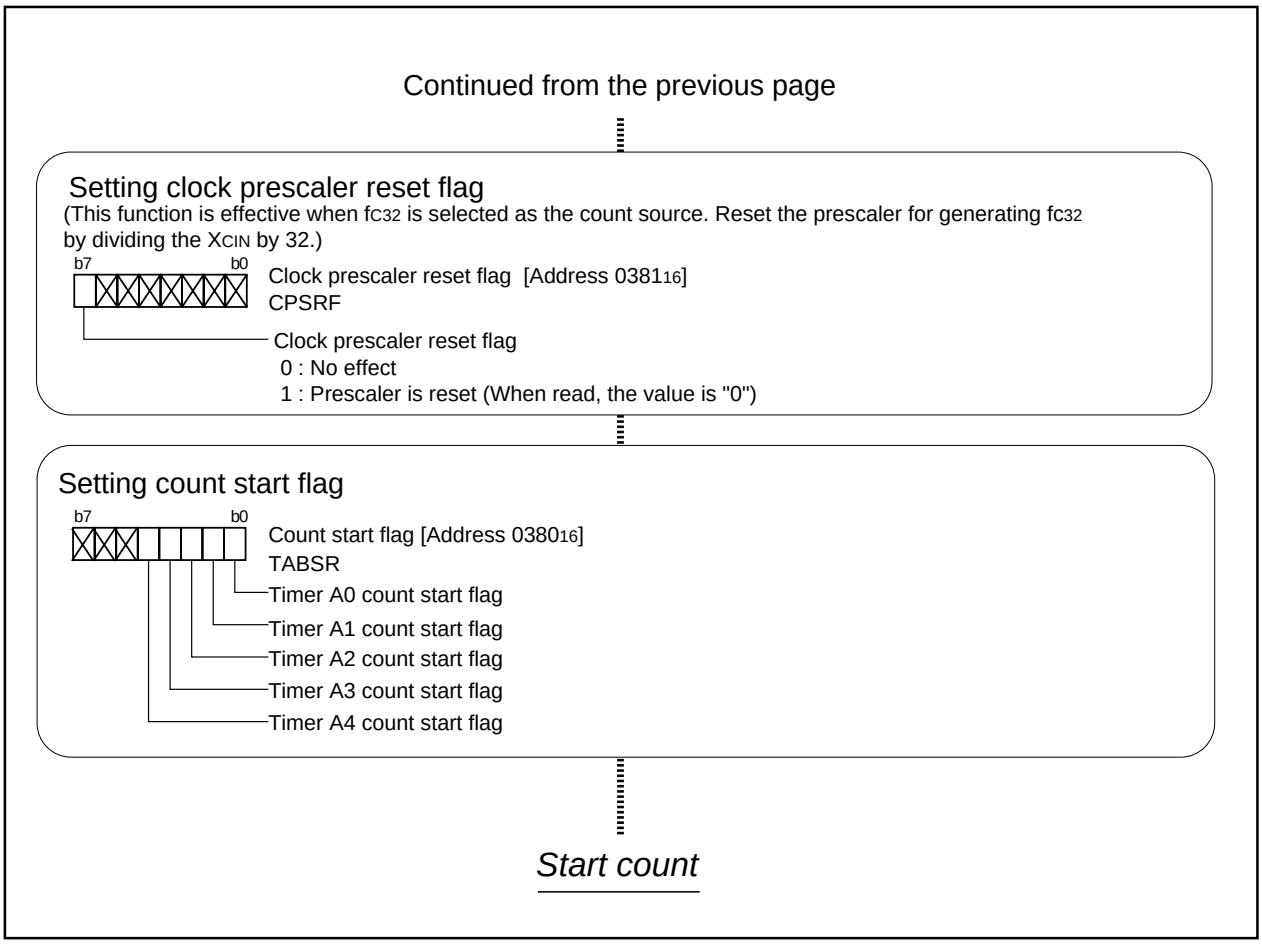

Continued from the previous page
Setting clock prescaler reset flag
(This function is effective when fC32 is selected as the count source. Reset the prescaler for generating fC32 by dividing the XCIN by 32.)
b7
b0
Clock prescaler reset flag [Address 038116]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")
Setting count start flag
b7
b0
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
Timer A1 count start flag
Timer A2 count start flag
Timer A3 count start flag
Timer A4 count start flag
Start count

Figure 2.2.27. Set-up procedure of pulse width modulation mode, 8-bit PWM mode selected (2)

2.2.12 Precautions for Timer A (timer mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer Ai register, then set the flag to “1”.

(2) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing shown in Figure 2.2.28 gets “FFFF16”. Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.

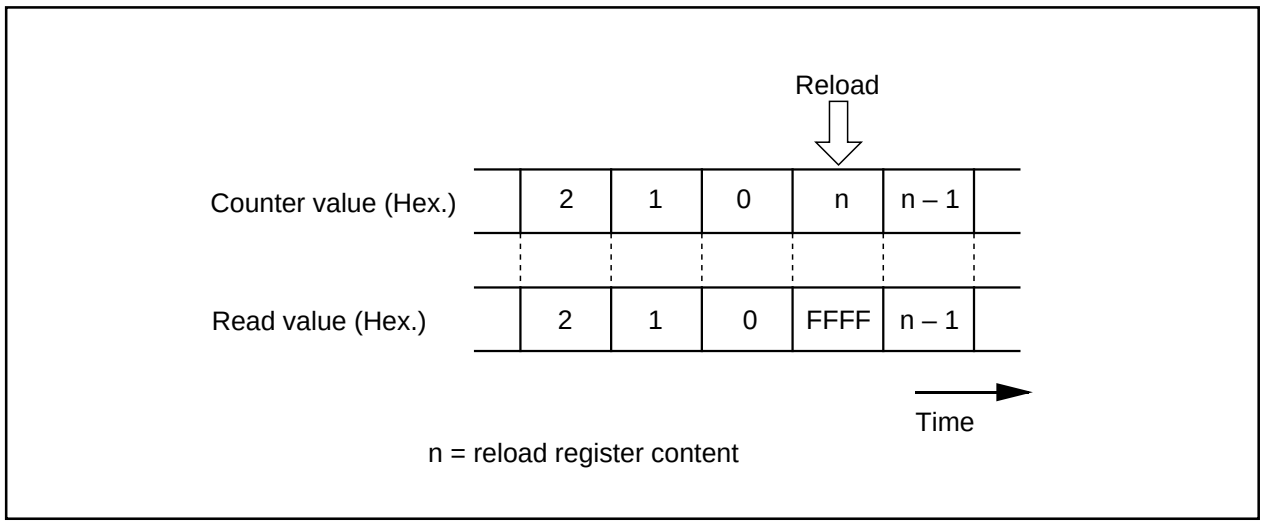


Figure 2.2.28. Reading timer Ai register

2.2.13 Precautions for Timer A (event counter mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer Ai register, then set the flag to “1”.

(2) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing shown in Figure 2.2.29 gets “FFFF16” by underflow or “000016” by overflow. Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.

(3) Please note the standards for the differences between the 2 pulses used in the two-phase pulse signals input signals to the TAiIN pin and TAiOUT pin (i = 2, 3, 4), as shown in Figure 2.2.30.

(4) When free run type is selected, if count is stopped, set a value in the timer Ai register again.

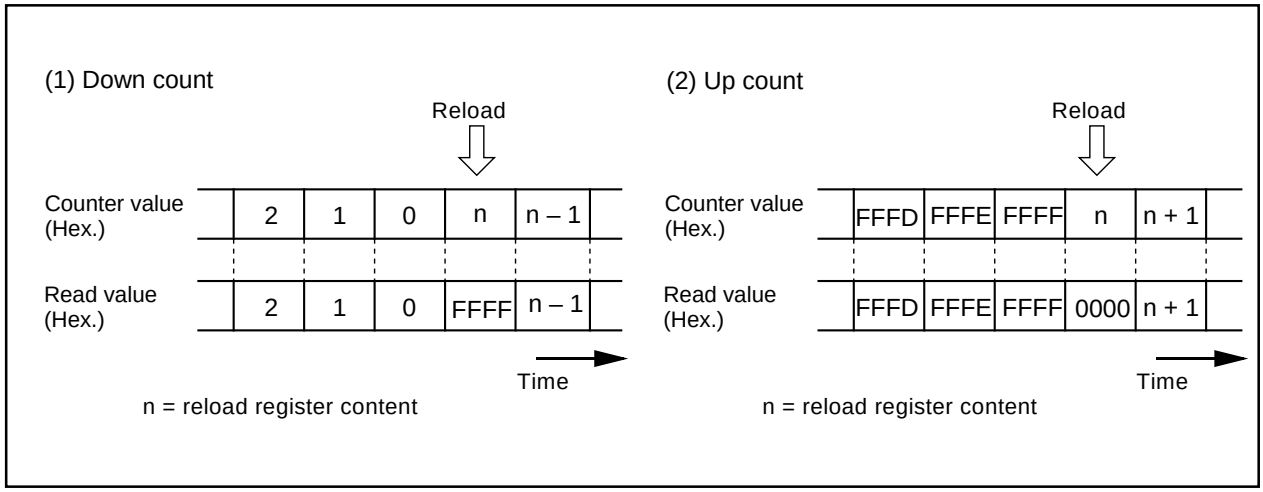


Figure 2.2.29. Reading timer Ai register

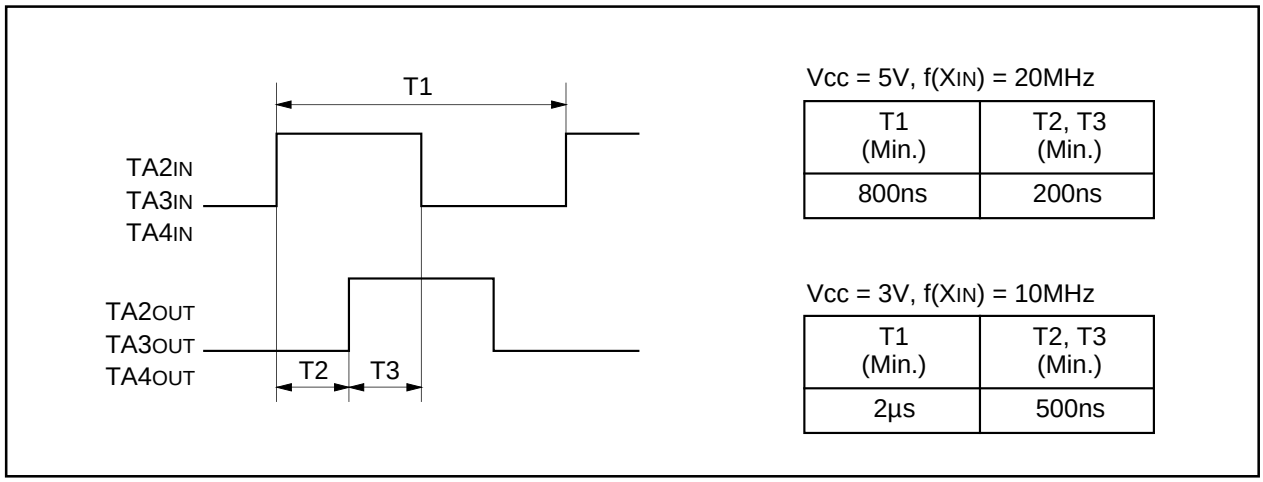


Figure 2.2.30. Standard of 2-phase pulses

(5) In the case of using as “Free-Run type”, the timer register contents may be unknown when counting begins. If the timer register is set before counting has started, then the starting value will be unknown.

- In the case where the up/down count will not be changed.

  Enable the “Reload” function and write to the timer register before counting begins. Rewrite the value to the timer register immediately after counting has started. If counting up, rewrite “0000_{16}” to the timer register. If counting down, rewrite “$FFFF_{16}$” to the timer register. This will cause the same operation as “Free-Run type” mode.

- In the case where the up/down count has changed.

  First set to “Reload type” operation. Once the first counting pulse has occurred, the timer may be changed to “Free-Run type”.

2.2.14 Precautions for Timer A (one-shot timer mode)

(1) At reset, the count start flag is set to “0”. Set a value in the timer Ai register, then set the flag to “1”.

(2) Setting the count start flag to “0” while a count is in progress causes as follows:
- The counter stops counting and a content of reload register is reloaded.
- The TAiOUT pin outputs “L” level.
- The interrupt request generated and the timer Ai interrupt request bit goes to “1”.

(3) The timer Ai interrupt request bit goes to “1” if the timer's operation mode is set using any of the following procedures:
- Selecting one-shot timer mode after reset.
- Changing operation mode from timer mode to one-shot timer mode.
- Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to “0” after the above listed changes have been made.

(4) If a trigger occurs while a count is in progress, after the counter performs one down count following the reoccurrence of a trigger, the reload register contents are reloaded, and the count continues. To generate a trigger while a count is in progress, generate the second trigger after an elapse longer than one cycle of the timer's count source after the previous trigger occurred.

2.2.15 Precautions for Timer A (pulse width modulation mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer Ai register, then set the flag to “1”.

(2) The timer Ai interrupt request bit becomes “1” if setting operation mode of the timer in compliance with any of the following procedures:

- Selecting PWM mode after reset.
- Changing operation mode from timer mode to PWM mode.
- Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to “0” after the above listed changes have been made.

(3) Setting the count start flag to “0” while PWM pulses are being output causes the counter to stop counting. If the TAiOUT pin is outputting an “H” level in this instance, the output level goes to “L”, and the timer Ai interrupt request bit goes to “1”. If the TAiOUT pin is outputting an “L” level in this instance, the level does not change, and the timer Ai interrupt request is not generated.

2.3 Clock-Synchronous Serial I/O

2.3.1 Overview

Clock-synchronous serial I/O carries out 8-bit data communications in synchronization with the clock. The following is an overview of the clock-synchronous serial I/O.

(1) Transmission/reception format

8-bit data

(2) Transfer rate

If the internal clock is selected as the transfer clock, the divide-by-2 frequency, resulting from the bit rate generator division, becomes the transfer rate. The bit rate generator count source can be selected from the following: f_1, f_8, and f_{32}. Clocks f_1, f_8, and f_{32} are derived by dividing the CPU's main clock by 1, 8, and 32 respectively.

Furthermore, if an external clock is selected as the transfer clock, the clock frequency input to the CLK pin becomes the transfer rate.

(3) Error detection

Only overrun errors can be detected. Overrun error is an error that occurs if the serial interface starts receiving the next data item before reading the contents of the UARTi receive buffer register and receives the 7th bit of the next data.

(4) How to deal with an error

- When receiving data, read an error flag and reception data simultaneously to determine which error has occurred. If the data read is erroneous, initialize the error flag and the UARTi receive buffer register, then receive the data again.

To initialize the UARTi receive buffer register

1. Set the receive enable bit to "0" (disable reception).
2. Set the serial I/O mode select bit to "000_2" (invalid serial I/O).
3. Set the serial I/O mode select bit.
4. Set the receive enable bit to "1" again (enable reception).

- To transmit data again due to an error such as staggered serial clock caused by noise, set the UARTi transmit buffer register again, then transmit the data again.

To set the UARTi transmit buffer register again

1. Set the serial I/O mode select bits to "000_2" (invalidate serial I/O).
2. Set the serial I/O mode select bits again.
3. Set the transmit enable bit to "1" (enable transmission), then set transmission data in the UARTi transmit buffer register.

(5) Function selection

For clock-synchronous serial I/O, the following functions can be selected:

(a) $\overline{\text{CTS}}/\overline{\text{RTS}}$ function

In the $\overline{\text{CTS}}$ function, an external IC can start transmission/reception by inputting an "L" level to the $\overline{\text{CTS}}$ pin. The $\overline{\text{CTS}}$ pin input level is detected when transmission/reception starts. Therefore, if the level is set to "H" during transmission/reception, it will stop from the next data.

The $\overline{\text{RTS}}$ function informs an external IC that $\overline{\text{RTS}}$ is reception-ready and has changed to "L". $\overline{\text{RTS}}$ goes back to "H" at the first falling edge of the transfer clock.

The clock-synchronous serial I/O has three types of $\overline{CTS}/\overline{RTS}$ functions to choose from:

- $\overline{CTS}/\overline{RTS}$ functions disabled — $\overline{CTS}/\overline{RTS}$ pin is a programmable I/O port.
- $\overline{CTS}$ function only enabled — $\overline{CTS}/\overline{RTS}$ pin performs the $\overline{CTS}$ function.
- $\overline{RTS}$ function only enabled — $\overline{CTS}/\overline{RTS}$ pin performs the $\overline{RTS}$ function.

(b) Function for choosing CLK polarity

This function switches the polarity of the transfer clock. The following operations are available:

- Data is input at the falling edge of the transfer clock, and is output at the rising edge.
- Data is input at the rising edge of the transfer clock, and is output at the falling edge.

(c) Function for choosing which bit to transmit/receive first

This function is to choose whether to transmit/receive data from bit 0 or from bit 7. Choose either of the following:

- LSB first — Data is transmitted/receivec from bit 0.
- MSB first — Data is transmitted/received from bit 7.

(d) Function for choosing continuous receive mode

Continuous receive mode is a mode in which reading the receive buffer register makes the reception-enabled status ready. In this mode, there is no need to write dummy data to the transmit buffer register so as to make the reception-enabled status ready. But at the time of starting reception, read the receive buffer register into a dummy manner.

- Normal mode — Writing dummy data to the transmit buffer register makes the reception enabled status ready.
- Continuous receive mode — Reading the reception buffer register makes the reception-enabled status ready.

(e) Data logic select function

This function is to reverse data when writing to transmit buffer register or reading from receive buffer register.

(f) Function for choosing a transmission interrupt factor

The timing to generate a transmission interrupt can be selected from the following: the instant the transmission buffer is emptied or the instant the transmission register is emptied. When transmission buffer empty timing is selected, an interrupt occurs when transmitted data is moved from the transmission buffer to the transmission register. Therefore, data can be transmitted in succession. When transmission register empty timing is selected, an interrupt occurs when data transmission is complete.

(g) TxD, RxD I/O polarity reverse function

This function is to reverse a polarity of TxD port output level and a polarity of RxD port input level.

Following are some examples in which various functions (a) through (g) are selected:

- Transmission Operation WITH: $\overline{CTS}$ function, transmission at falling edge of transfer clock, LSB First, interrupt at instant transmission buffer is emptied
- Transmission Operation WITH: $\overline{CTS}/\overline{RTS}$ function disabled, transmission at falling edge of transfer clock, LSB First, interrupt at instant transmission is completed
- Reception Operation WITH: $\overline{RTS}$ function, reception at falling edge of transfer clock, LSB First, suc-

cessive reception mode disabled

(6) Input to the serial I/O and the direction register

To input an external signal to the serial I/O, slect the function select register A to I/O port and set the direction register to input.

(7) Pins related to the serial I/O

- $\overline{CTS_0}$, $\overline{CTS_1}$, $\overline{CTS_2}$, $\overline{CTS_3}$ pins Input pins for the $\overline{CTS}$ function
- $\overline{RTS_0}$, $\overline{RTS_1}$, $\overline{RTS_2}$, $\overline{RTS_3}$ pins Output pins for the $\overline{RTS}$ function
- CLK_0, CLK_1, CLK_2, CLK_3 pins Input/output pins for the transfer clock
- RxD_0, RxD_1, RxD_2, RxD_3 pins Input pins for data
- TxD_0, TxD_1, TxD_2, TxD_3 pins Output pins for data (Note)

Note: Since TxD_2 pin is N-channel open drain, this pin needs pull-up resistor.

(8) Registers related to the serial I/O

Figure 2.3.1 shows the memory map of serial I/O-related registers, and Figures 2.3.2 to 2.3.4 show serial I/O-related registers.

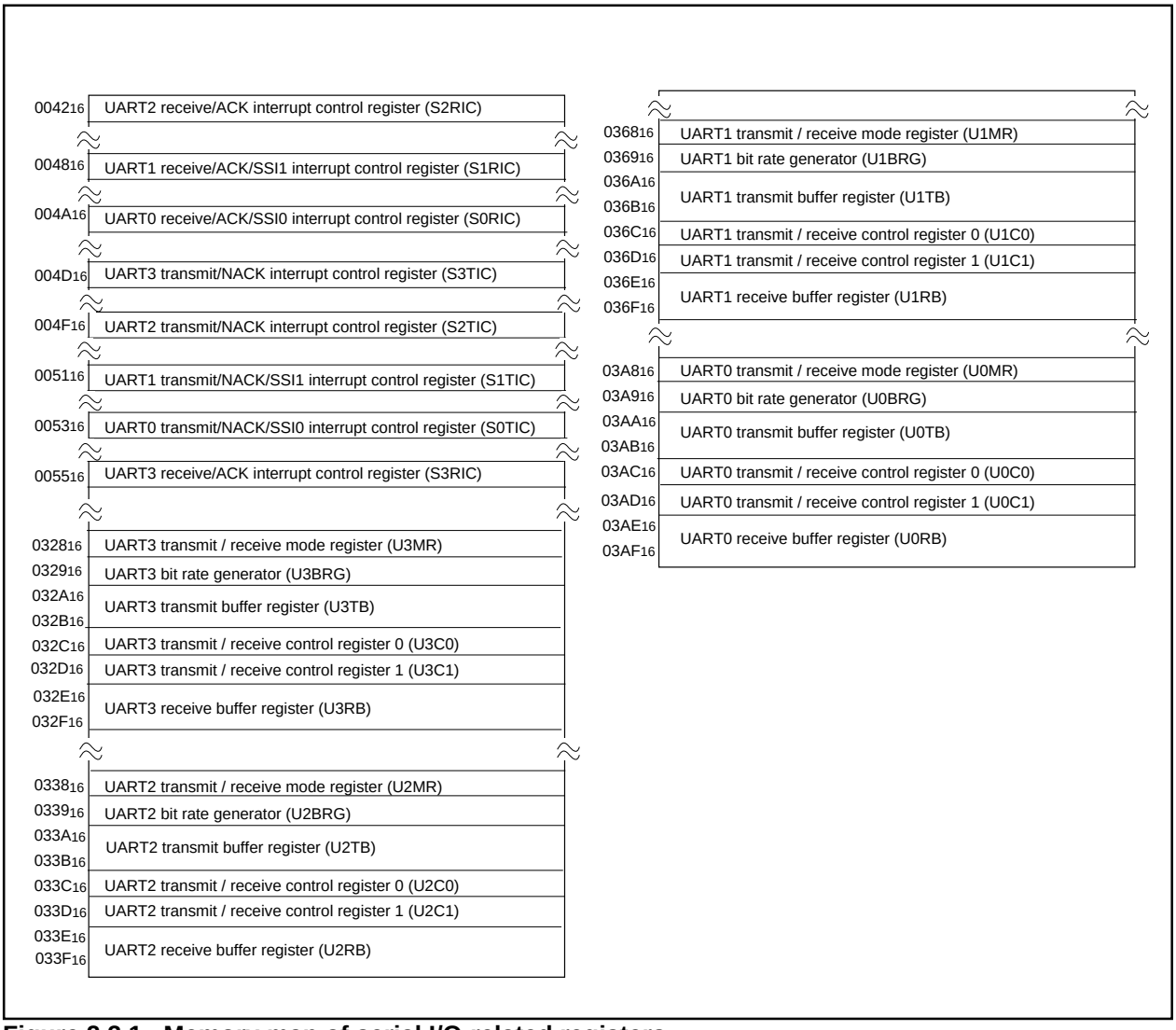
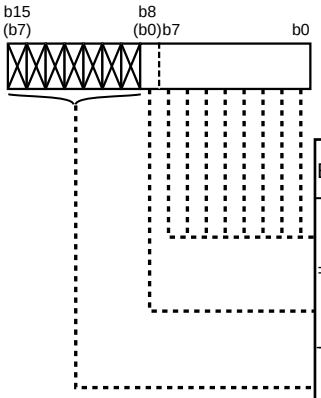

0042₁₆ UART2 receive/ACK interrupt control register (S2RIC)
0048₁₆ UART1 receive/ACK/SSI1 interrupt control register (S1RIC)
004A₁₆ UART0 receive/ACK/SSI0 interrupt control register (S0RIC)
004D₁₆ UART3 transmit/NACK interrupt control register (S3TIC)
004F₁₆ UART2 transmit/NACK interrupt control register (S2TIC)
0051₁₆ UART1 transmit/NACK/SSI1 interrupt control register (S1TIC)
0053₁₆ UART0 transmit/NACK/SSI0 interrupt control register (S0TIC)
0055₁₆ UART3 receive/ACK interrupt control register (S3RIC)
0328₁₆ UART3 transmit / receive mode register (U3MR)
0329₁₆ UART3 bit rate generator (U3BRG)
032A₁₆ 032B₁₆ UART3 transmit buffer register (U3TB)
032C₁₆ UART3 transmit / receive control register 0 (U3C0)
032D₁₆ UART3 transmit / receive control register 1 (U3C1)
032E₁₆ 032F₁₆ UART3 receive buffer register (U3RB)
0338₁₆ UART2 transmit / receive mode register (U2MR)
0339₁₆ UART2 bit rate generator (U2BRG)
033A₁₆ 033B₁₆ UART2 transmit buffer register (U2TB)
033C₁₆ UART2 transmit / receive control register 0 (U2C0)
033D₁₆ UART2 transmit / receive control register 1 (U2C1)
033E₁₆ 033F₁₆ UART2 receive buffer register (U2RB)
0368₁₆ UART1 transmit / receive mode register (U1MR)
0369₁₆ UART1 bit rate generator (U1BRG)
036A₁₆ 036B₁₆ UART1 transmit buffer register (U1TB)
036C₁₆ UART1 transmit / receive control register 0 (U1C0)
036D₁₆ UART1 transmit / receive control register 1 (U1C1)
036E₁₆ 036F₁₆ UART1 receive buffer register (U1RB)
03A8₁₆ UART0 transmit / receive mode register (U0MR)
03A9₁₆ UART0 bit rate generator (U0BRG)
03AA₁₆ 03AB₁₆ UART0 transmit buffer register (U0TB)
03AC₁₆ UART0 transmit / receive control register 0 (U0C0)
03AD₁₆ UART0 transmit / receive control register 1 (U0C1)
03AE₁₆ 03AF₁₆ UART0 receive buffer register (U0RB)

Figure 2.3.1. Memory map of serial I/O-related registers

UARTi transmit buffer register (i= 0 to 3) (Note)

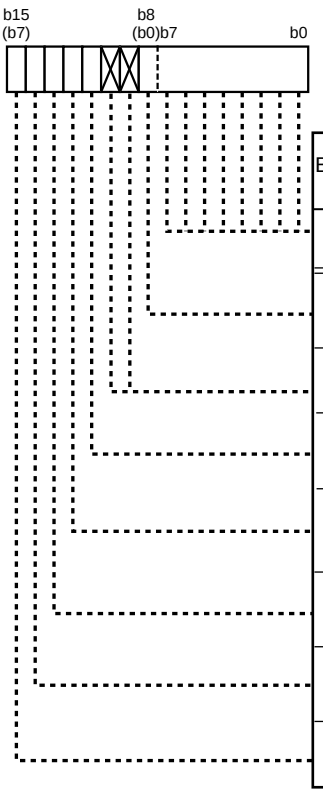


Symbol	Address	When reset
U0TB	$03AB_{16}$, $03AA_{16}$	Indeterminate
U1TB	$036B_{16}$, $036A_{16}$	Indeterminate
U2TB	$033B_{16}$, $033A_{16}$	Indeterminate
U3TB	$032B_{16}$, $032A_{16}$	Indeterminate

Bit Symbol	Function (clock synchronous serial I/O mode)	Function (UART mode)	R	W
——	Transmit data	Transmit data	×	○
——	——	Transmit data (9th bit)	×	○
——	Nothing is assigned. Write “0” when writing to these bits. The values are indeterminate when read.		–	–

Note: Use MOV instruction to write to this register.

UARTi receive buffer register (i= 0 to 3)



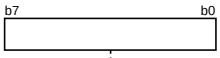
Symbol	Address	When reset
U0RB	$03AF_{16}$, $03AE_{16}$	Indeterminate
U1RB	$036F_{16}$, $036E_{16}$	Indeterminate
U2RB	$033F_{16}$, $033E_{16}$	Indeterminate
U3RB	$032F_{16}$, $032E_{16}$	Indeterminate

Bit Symbol	Bit Name	Function (clock synchronous serial I/O mode)	Function (UART mode)	R	W
——	——	Receive data	Receive data	○	×
——	——	——	Receive data (9th bit)	○	×
——	Nothing is assigned. Write “0” when writing to these bits. The values are indeterminate when read.			–	–
ABT	Arbitration lost detecting flag (Note 1)	0 : Not detected 1 : Detected	Invalid	○	○
OER	Overrun error flag (Note 2)	0 : No overrun error 1 : Overrun error	0 : No overrun error 1 : Overrun error	○	×
FER	Framing error flag (Note 2)	Invalid	0 : No framing error 1 : Framing error	○	×
PER	Parity error flag (Note 2)	Invalid	0 : No parity error 1 : Parity error	○	×
SUM	Error sum flag (Note 2)	Invalid	0 : No error 1 : Error	○	×

Note 1: Always write “0”.
Note 2: Bits 15 to 12 are set to “0000_2” when the serial I/O mode select bit (bits 0 to 2 at addresses $03A8_{16}$, 0368_{16}, 0338_{16}, 0328_{16}) are set to “000_2” or the receive enable bit is set to “0”. Bit 15 is set to “0” when all of bits 14 to 12 are set to “0”.
Bits 14 and 13 are also set to “0” when the lower byte of the UARTi receive buffer register (addresses $03AE_{16}$, $036E_{16}$, $033E_{16}$, $032E_{16}$) is read.

Figure 2.3.2. Serial I/O-related registers (1)

UARTi bit rate generator (o=0 to 3) (Note 1, 2)

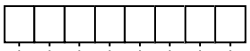


Symbol	Address	When reset
U0BRG	$03A9_{16}$	Indeterminate
U1BRG	0369_{16}	Indeterminate
U2BRG	0339_{16}	Indeterminate
U3BRG	0329_{16}	Indeterminate

Function	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}	–	○

Note 1: Use MOV instruction to write to this register.
Note 2: Write a value to this register while transmit/receive halts.

UARTi transmit/receive mode register (i = 0 to 3)



Symbol	Address	When reset
U0MR	$03A8_{16}$	00_{16}
U1MR	0368_{16}	00_{16}
U2MR	0338_{16}	00_{16}
U3MR	0328_{16}	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit (Note 3)	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Serial I/O mode 0 1 1 : I^2C mode Inhibited except in cases listed above	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long Inhibited except in cases listed above	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 1)	0 : Internal clock 1 : External clock (Note 1)	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	TxD, RxD input/output polarity switch bit (Note 2)	0 : Normal 1 : Reversed		○	○

Note 1: When I^2C bus interface mode is selected, set the port direction register for the corresponding port (SCLi) to 0, or the port direction register to 1 and the port data register to 1. When a mode other than serial I/O mode is selected, set the port direction register for the corresponding port (CLKi) to 0.
Note 2: Normally set to “0”.
Note 3: Set the RxDi pin's port direction register to “0” when receiving.

Figure 2.3.3. Serial I/O-related registers (2)

UARTi transmit/receive control register 0 (i=0 to 3)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiC0 (i=0 to 3)	03AC16, 036C16, 033C16, 032C16,	0816

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f1 is selected 0 1 : f8 is selected 1 0 : f32 is selected 1 1 : Inhibited	b1 b0 **0 0 : f1 is selected** **0 1 : f8 is selected** **1 0 : f32 is selected** **1 1 : Inhibited**	○	○
CLK1				○	○
CRS	CTS/RTS function select bit	Valid when bit 4 = "0" 0 : CTS function is selected (Note 1) 1 : RTS function is selected (Note 4)	Valid when bit 4 = "0" 0 : CTS function is selected (Note 1) 1 : RTS function is selected (Note 4)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	CTS/RTS disable bit	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled	○	○
NCH (Note 2)	Data output select bit	0 : TxDi/SDAi and SCLi pin is CMOS output 1 : TxDi/SDAi and SCLi pin is N-channel open drain output	0 : TxDi/SDAi and SCLi pin is CMOS output 1 : TxDi/SDAi and SCLi pin is N-channel open drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Set to "0"	○	○
UFORM	Transfer format select bit (Note 3)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

Note 1: Set the corresponding port direction register to "0".
Note 2: UART2 transfer pin (TxD2: P70 and SCL2: P71) is N-channel open drain output. It cannot be set to CMOS output.
Note 3: Only clock synchronous serial I/O mode and 8-bit UART mode are valid.
Note 4: The corresponding port register and port direction register are invalid.

UARTi transmit/receive control register 1 (i= 0 to 3)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiC1 (i=0 to 3)	03AD16, 036D16, 033D16, 032D16	0216

Bit Symbol	Bit Name	Function (clock synchronous serial I/O mode)	Function (UART mode)	R	W
TE	Transmit enable bit	0 : Transmit disabled 1 : Transmit enabled		○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register		○	○
RE	Receive enable bit	0 : Receive disabled 1 : Receive enabled		○	○
RI	Receive complete flag	0 : Data packet in receive buffer register 1 : No data packet in receive buffer register		○	○
UiIRS	UARTi transmit interrupt cause select bit	0 : Transmit buffer empty (TI =1) 1 : Transmit buffer completed (TXEPT =1)		○	○
UiRRM	UARTi continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Set to "0"	○	○
UiLCH	Data logic select bit	0 : No reverse 1 : Reverse		○	○
UiERE	Error signal output enable bit	Set to "0" The value is indeterminate when read.	0 : Output disabled 1 : Output enabled (Note 1)	○	○

Note 1: When disabling the error signal output, set the UiERE bit to "0" after setting the UiMR register.

Figure 2.3.4. Serial I/O-related registers (3)

2.3.2 Operation of Serial I/O (transmission in clock-synchronous serial I/O mode)

In transmitting data in clock-synchronous serial I/O mode, choose functions from those listed in Table 2.3.1. Operations of the circled items are described below. Figure 2.3.5 shows the operation timing, and Figures 2.3.6 and 2.3.7 show the set-up procedures.

Table 2.3.1. Choosed functions

Item	Set-up		Item	Set-up	
Transfer clock source	O	Internal clock (f_1 / f_8 / f_{32})	Transfer clock	O	LSB first
		External clock (CLKi pin)			MSB first
$\overline{CTS}$ function	O	$\overline{CTS}$ function enabled	Transmission interrupt factor	O	Transmission buffer empty
		$\overline{CTS}$ function disabled			Transmission complete
CLK polarity	O	Output transmission data at the falling edge of the transfer clock	Data logic select function	O	No reverse
					Reverse
		Output transmission data at the rising edge of the transfer clock	TxD, RxD I/O polarity reverse bit	O	No reverse
					Reverse

Operation

(1) Setting the transmit enable bit to “1” and writing transmission data to the UARTi transmit buffer register makes data transmissible status ready.

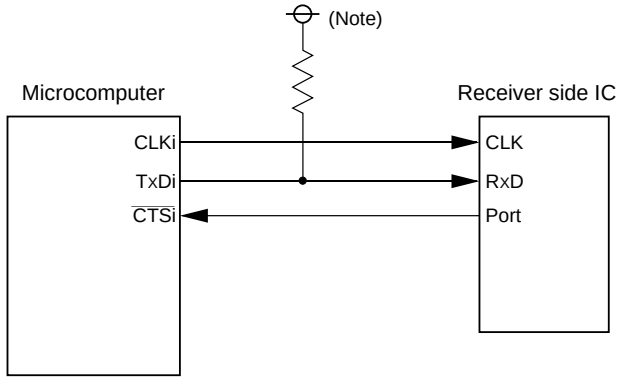
(2) When input to the $\overline{CTS}$i pin goes to “L” level, transmission starts (the $\overline{CTS}$i pin must be controlled on the reception side).

(3) In synchronization with the first falling edge of the transfer clock, transmission data held in the UARTi transmit buffer register is transmitted to the UARTi transmit register. At this time, the UARTi transmit interrupt request bit goes to “1”. Also, the first bit of the transmission data is transmitted from the TxDi pin. Then the data is transmitted bit by bit from the lower order in synchronization with the falling edges.

(4) When transmission of 1-byte data is completed, the transmit register empty flag goes to “1”, which indicates that transmission is completed. The transfer clock stops at “H” level.

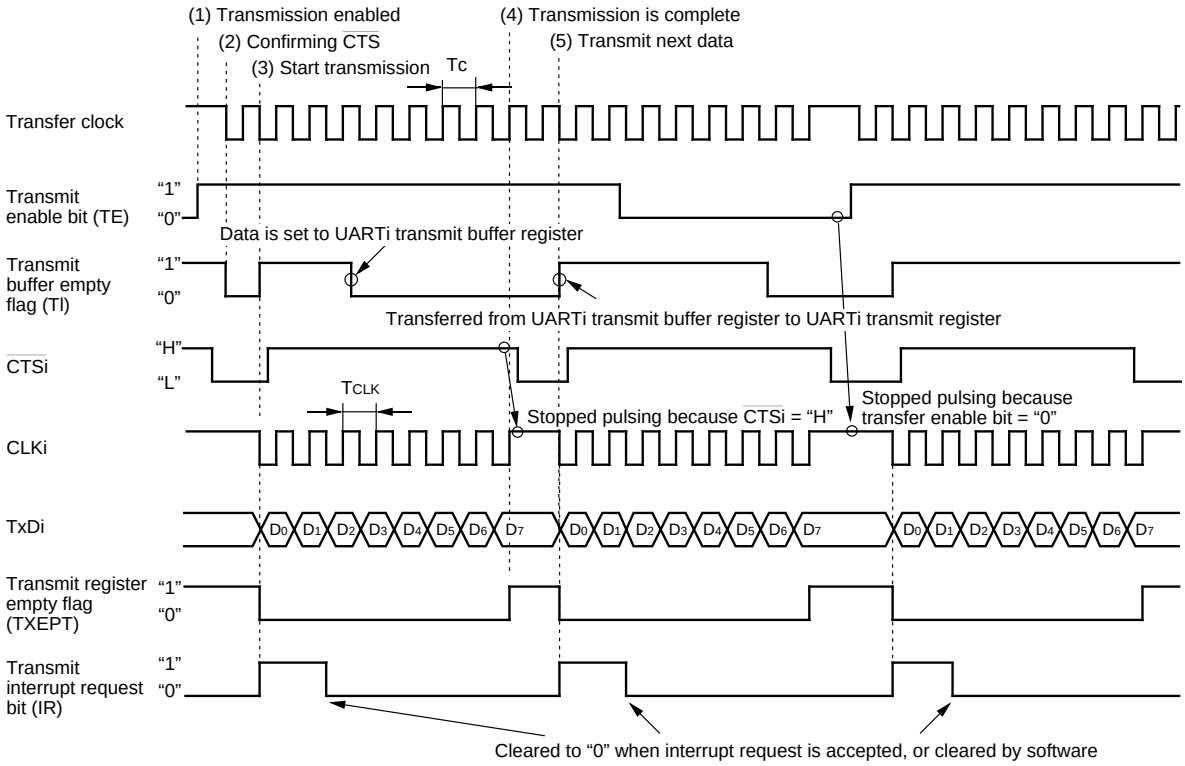
(5) If the next transmission data is set in the UARTi transmit buffer register while transmission is in progress (before the eighth bit has been transmitted), the data is transmitted in succession.

Example of wiring



Note : Since TxD2 pin is N-channel open drain, this pin needs pull-up resistance.

Example of operation


(1) Transmission enabled
(2) Confirming CTS
(3) Start transmission
Tc
(4) Transmission is complete
(5) Transmit next data
Transfer clock
Transmit enable bit (TE)
"1"
"0"
Data is set to UARTi transmit buffer register
Transmit buffer empty flag (TI)
Transferred from UARTi transmit buffer register to UARTi transmit register
CTSi
"H"
"L"
TCLK
Stopped pulsing because CTSi = "H"
Stopped pulsing because transfer enable bit = "0"
CLKi
TxDi
D0 D1 D2 D3 D4 D5 D6 D7
Transmit register empty flag (TXEPT)
Transmit interrupt request bit (IR)
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.
The above timing applies to the following settings:
- Internal clock is selected.
- CTS function is selected.
- CLK polarity select bit = "0".
- Transmit interrupt cause select bit = "0".

Tc = TCLK = 2(n + 1) / fi
fi: frequency of BRGi count source (f1, f8, f32)
n: value set to BRGi

Figure 2.3.5. Operation timing of transmission in clock-synchronous serial I/O mode

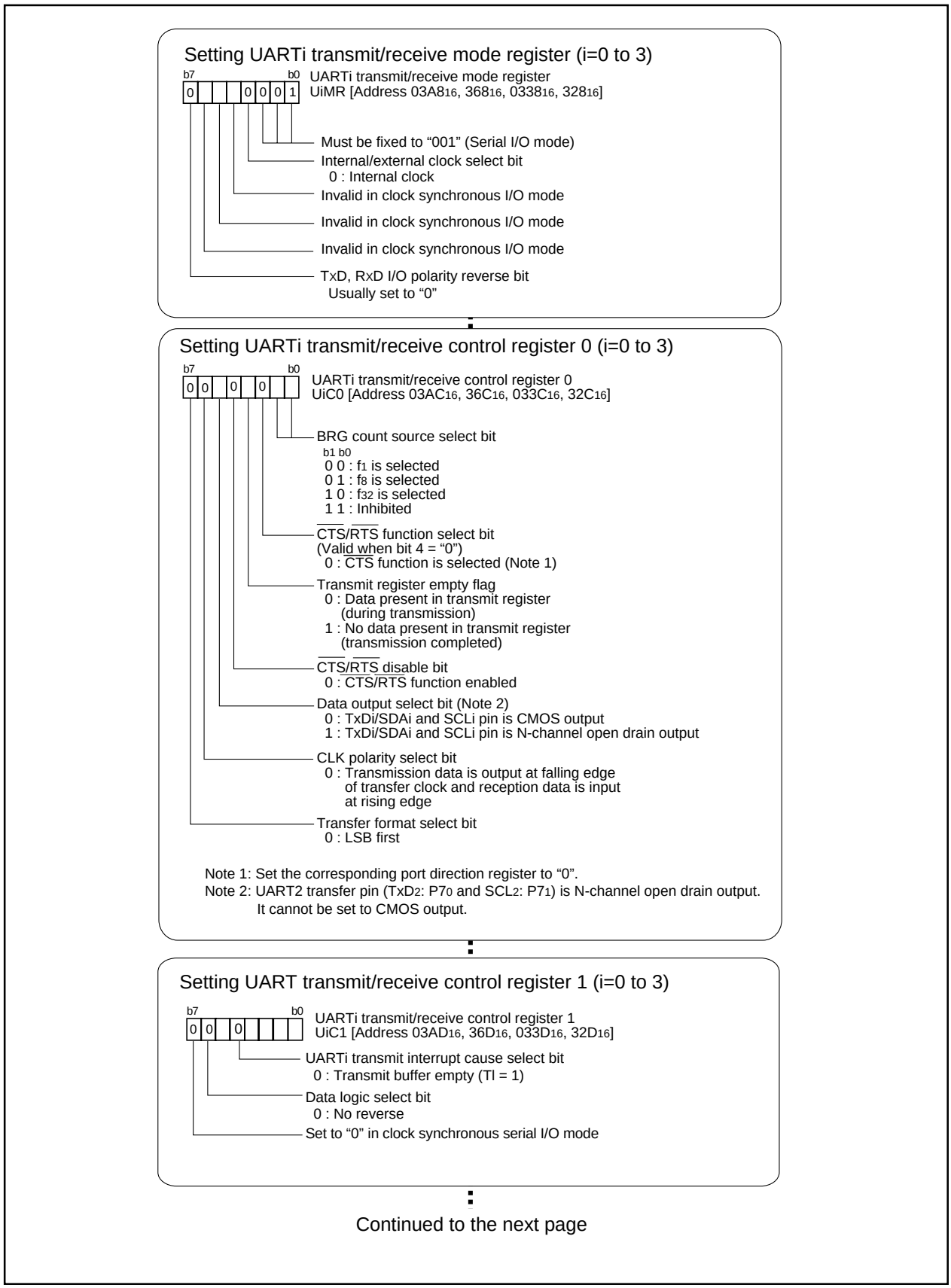

Setting UARTi transmit/receive mode register (i=0 to 3)
b7 b0
0 0 0 0 0 1
UARTi transmit/receive mode register
UiMR [Address 03A816, 36816, 033816, 32816]
Must be fixed to "001" (Serial I/O mode)
Internal/external clock select bit
0 : Internal clock
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
TxD, RxD I/O polarity reverse bit
Usually set to "0"
Setting UARTi transmit/receive control register 0 (i=0 to 3)
b7 b0
0 0 0 0
UARTi transmit/receive control register 0
UiC0 [Address 03AC16, 36C16, 033C16, 32C16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : Inhibited
CTS/RTS function select bit
(Valid when bit 4 = "0")
0 : CTS function is selected (Note 1)
Transmit register empty flag
0 : Data present in transmit register
(during transmission)
1 : No data present in transmit register
(transmission completed)
CTS/RTS disable bit
0 : CTS/RTS function enabled
Data output select bit (Note 2)
0 : TxDi/SDAi and SCLi pin is CMOS output
1 : TxDi/SDAi and SCLi pin is N-channel open drain output
CLK polarity select bit
0 : Transmission data is output at falling edge
of transfer clock and reception data is input
at rising edge
Transfer format select bit
0 : LSB first
Note 1: Set the corresponding port direction register to "0".
Note 2: UART2 transfer pin (TxD2: P70 and SCL2: P71) is N-channel open drain output.
It cannot be set to CMOS output.
Setting UART transmit/receive control register 1 (i=0 to 3)
b7 b0
0 0 0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
UARTi transmit interrupt cause select bit
0 : Transmit buffer empty (TI = 1)
Data logic select bit
0 : No reverse
Set to "0" in clock synchronous serial I/O mode
Continued to the next page

Figure 2.3.6. Set-up procedure of transmission in clock-synchronous serial I/O mode (1)

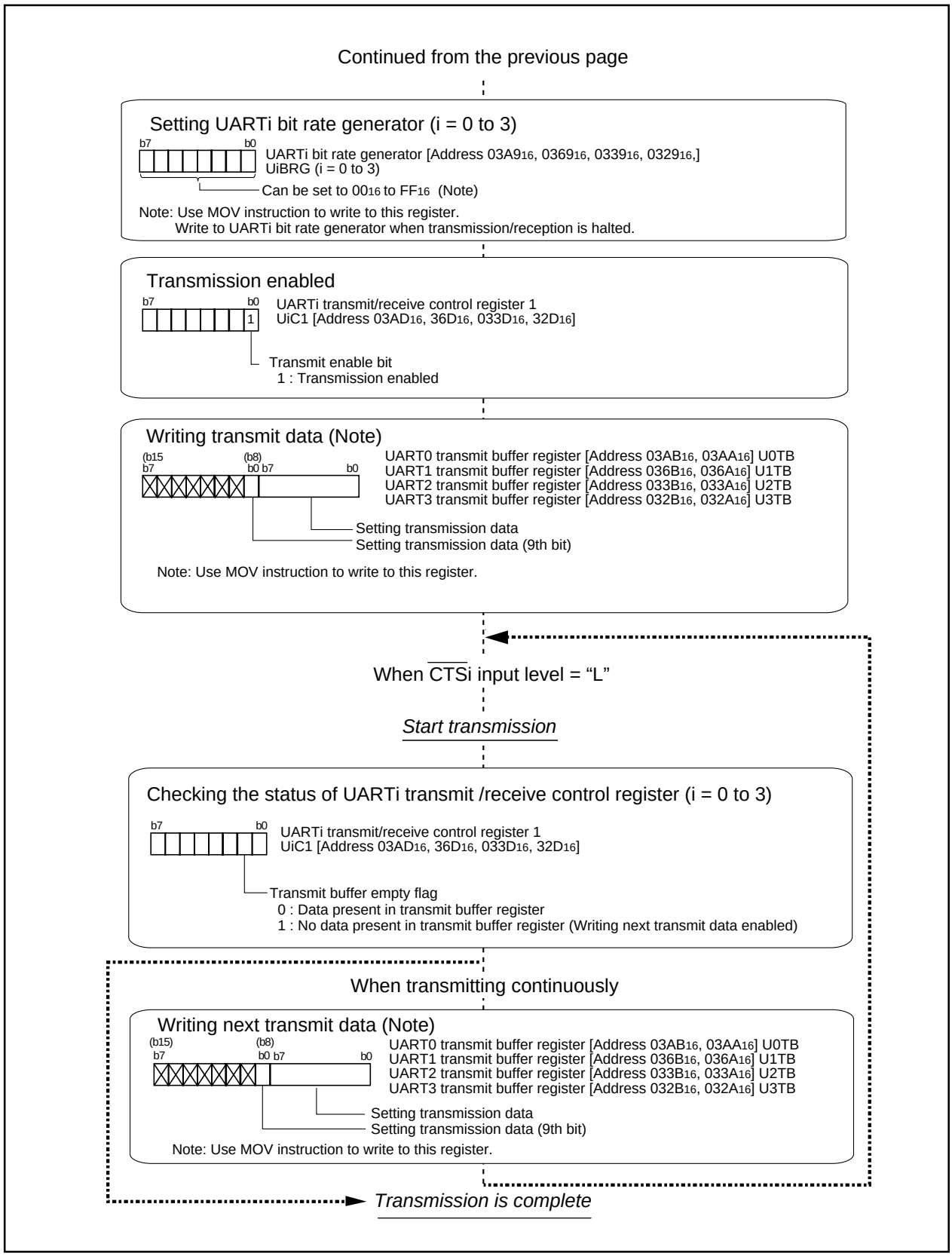

Continued from the previous page
Setting UARTi bit rate generator (i = 0 to 3)
b7 b0
UARTi bit rate generator [Address 03A916, 036916, 033916, 032916,]
UiBRG (i = 0 to 3)
Can be set to 0016 to FF16 (Note)
Note: Use MOV instruction to write to this register.
Write to UARTi bit rate generator when transmission/reception is halted.
Transmission enabled
b7 b0
1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit enable bit
1 : Transmission enabled
Writing transmit data (Note)
(b15 (b8)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting transmission data
Setting transmission data (9th bit)
Note: Use MOV instruction to write to this register.
When CTSi input level = "L"
Start transmission
Checking the status of UARTi transmit /receive control register (i = 0 to 3)
b7 b0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit buffer empty flag
0 : Data present in transmit buffer register
1 : No data present in transmit buffer register (Writing next transmit data enabled)
When transmitting continuously
Writing next transmit data (Note)
(b15) (b8)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting transmission data
Setting transmission data (9th bit)
Note: Use MOV instruction to write to this register.
Transmission is complete

Figure 2.3.7. Set-up procedure of transmission in clock-synchronous serial I/O mode (2)

2.3.3 Operation of Serial I/O (reception in clock-synchronous serial I/O mode)

In receiving data in clock-synchronous serial I/O mode, choose functions from those listed in Table 2.3.2. Operations of the circled items are described below. Figure 2.3.8 shows the operation timing, and Figures 2.3.9 and 2.3.10 show the set-up procedures.

Table 2.3.2. Choosed functions

Item	Set-up		Item	Set-up	
Transfer clock source	O	Internal clock (f_1 / f_8 / f_{32})	Transfer clock	O	LSB first
		External clock (CLKi pin)			MSB first
$\overline{RTS}$ function	O	$\overline{RTS}$ function enabled	Continuous receive mode	O	Disabled
		$\overline{RTS}$ function disabled			Enabled
CLK polarity	O	Output transmission data at the falling edge of the transfer clock	Data logic select function	O	No reverse
					Reverse
		Output transmission data at the rising edge of the transfer clock	TxD, RxD I/O polarity reverse bit	O	No reverse
					Reverse

Operation

(1) Writing dummy data to the UARTi transmit buffer register, setting the receive enable bit to “1”, and the transmit enable bit to “1”, makes the data receivable status ready. At this time, the output from the $\overline{RTSi}$ pin goes to “L” level, which informs the transmission side that the data receivable status is ready (output the transfer clock from the IC on the transmission side after checking that the $\overline{RTS}$ output has gone to “L” level).

(2) In synchronization with the first rising edge of the transfer clock, the input signal to the RxDi pin is stored in the highest bit of the UARTi receive register. Then, data is taken in by shifting right the content of the UARTi reception data in synchronization with the rising edges of the transfer clock.

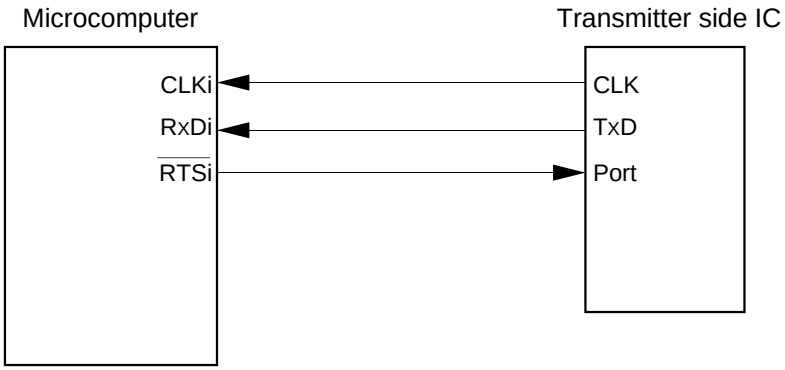
(3) When 1-byte data lines up in the UARTi receive register, the content of the UARTi receive register is transmitted to the UARTi receive buffer register. The transfer clock stops at “H” level. At this time, the receive complete flag and the UARTi receive interrupt request bit goes to “1”.

(4) The receive complete flag goes to “0” when the lower-order byte of the UARTi buffer register is read.

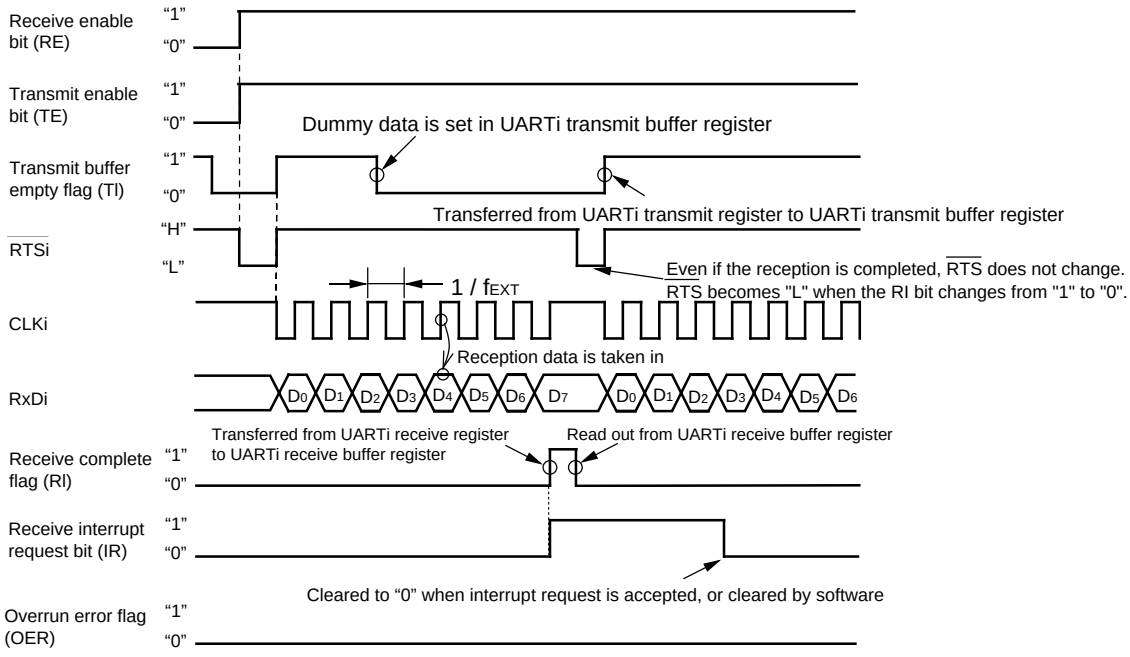
Note

• Set CLKi and RxDi pins' port direction register to “0”.

Example of wiring



Example of operation


Receive enable bit (RE)
Transmit enable bit (TE)
Transmit buffer empty flag (TI)
RTSi
CLKi
RxDi
Receive complete flag (RI)
Receive interrupt request bit (IR)
Overrun error flag (OER)
"1"
"0"
"H"
"L"
Dummy data is set in UARTi transmit buffer register
Transferred from UARTi transmit register to UARTi transmit buffer register
Even if the reception is completed, RTS does not change.
RTS becomes "L" when the RI bit changes from "1" to "0".
1 / fEXT
Reception data is taken in
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6
Transferred from UARTi receive register to UARTi receive buffer register
Read out from UARTi receive buffer register
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- External clock is selected.
- RTS function is selected.
- CLK polarity select bit = "0".

fEXT: frequency of external clock

Make sure that the following conditions are met when the CLKi pin input ="H" before data reception
- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UARTi transmit buffer register

Figure 2.3.8. Operation timing of reception in clock-synchronous serial I/O mode

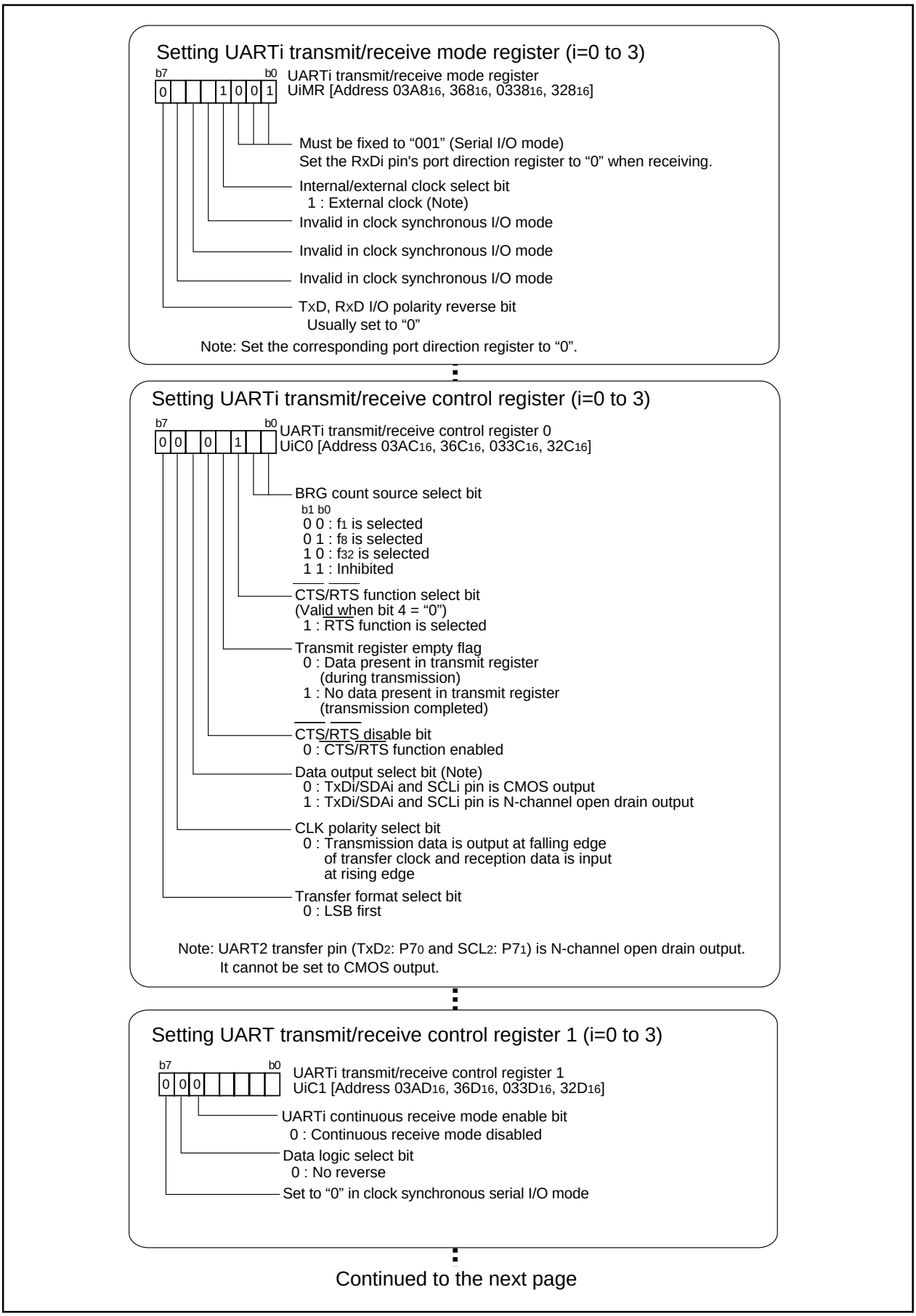

Setting UARTi transmit/receive mode register (i=0 to 3)
b7 b0
0 1 0 0 1
UARTi transmit/receive mode register
UiMR [Address 03A816, 36816, 033816, 32816]
Must be fixed to "001" (Serial I/O mode)
Set the RxDi pin's port direction register to "0" when receiving.
Internal/external clock select bit
1 : External clock (Note)
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
TxD, RxD I/O polarity reverse bit
Usually set to "0"
Note: Set the corresponding port direction register to "0".
Setting UARTi transmit/receive control register (i=0 to 3)
b7 b0
0 0 0 0 1
UARTi transmit/receive control register 0
UiC0 [Address 03AC16, 36C16, 033C16, 32C16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : Inhibited
CTS/RTS function select bit
(Valid when bit 4 = "0")
1 : RTS function is selected
Transmit register empty flag
0 : Data present in transmit register
(during transmission)
1 : No data present in transmit register
(transmission completed)
CTS/RTS disable bit
0 : CTS/RTS function enabled
Data output select bit (Note)
0 : TxDi/SDAi and SCLi pin is CMOS output
1 : TxDi/SDAi and SCLi pin is N-channel open drain output
CLK polarity select bit
0 : Transmission data is output at falling edge
of transfer clock and reception data is input
at rising edge
Transfer format select bit
0 : LSB first
Note: UART2 transfer pin (TxD2: P70 and SCL2: P71) is N-channel open drain output.
It cannot be set to CMOS output.
Setting UART transmit/receive control register 1 (i=0 to 3)
b7 b0
0 0 0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
UARTi continuous receive mode enable bit
0 : Continuous receive mode disabled
Data logic select bit
0 : No reverse
Set to "0" in clock synchronous serial I/O mode
Continued to the next page

Figure 2.3.9. Set-up procedure of reception in clock-synchronous serial I/O mode (1)

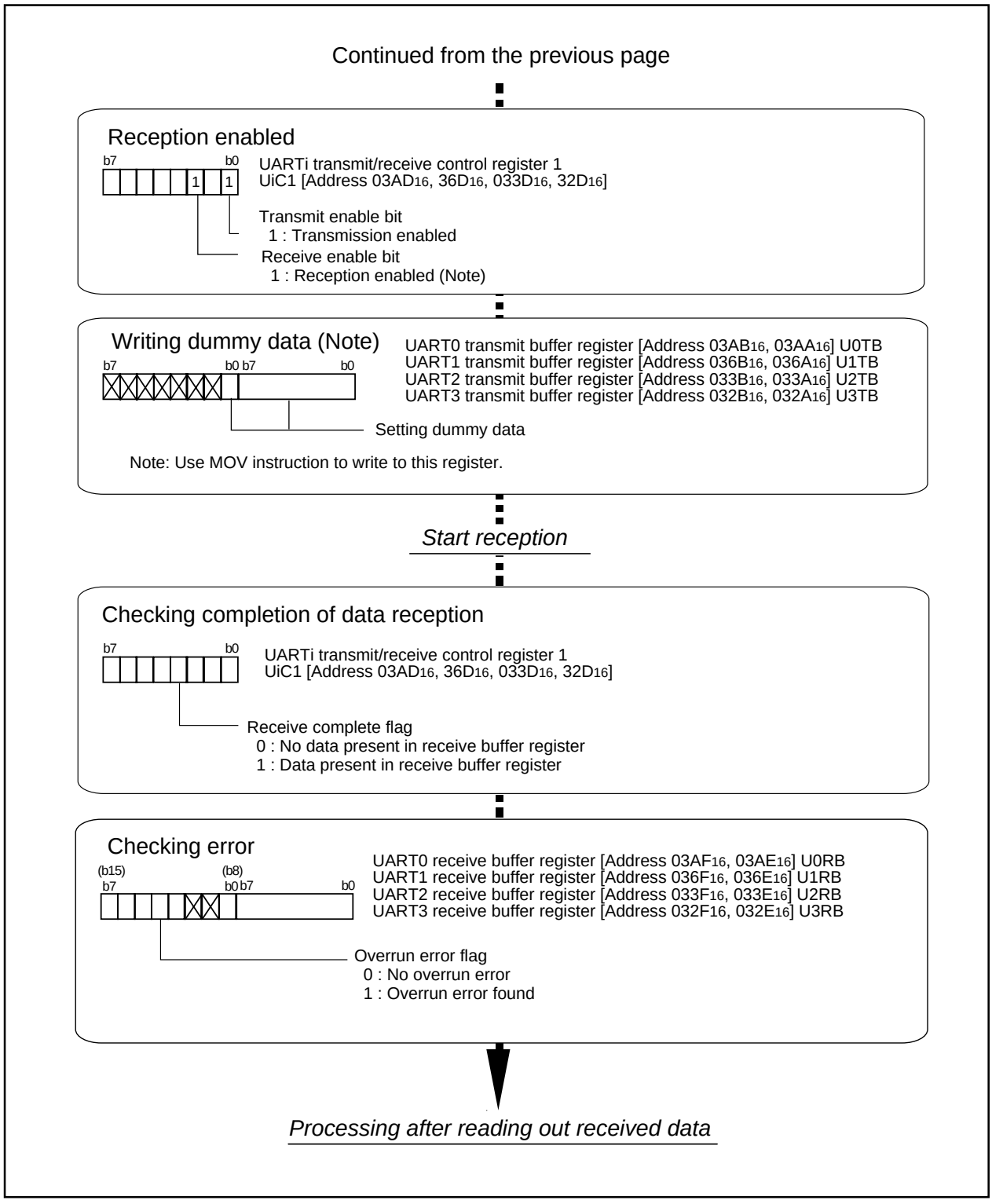

Continued from the previous page
Reception enabled
b7 b0
1 1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit enable bit
1 : Transmission enabled
Receive enable bit
1 : Reception enabled (Note)
Writing dummy data (Note)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting dummy data
Note: Use MOV instruction to write to this register.
Start reception
Checking completion of data reception
b7 b0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Receive complete flag
0 : No data present in receive buffer register
1 : Data present in receive buffer register
Checking error
(b15) (b8)
b7 b0 b7 b0
UART0 receive buffer register [Address 03AF16, 03AE16] U0RB
UART1 receive buffer register [Address 036F16, 036E16] U1RB
UART2 receive buffer register [Address 033F16, 033E16] U2RB
UART3 receive buffer register [Address 032F16, 032E16] U3RB
Overrun error flag
0 : No overrun error
1 : Overrun error found
Processing after reading out received data

Figure 2.3.10. Set-up procedure of reception in clock-synchronous serial I/O mode (2)

2.3.4 Precautions for Serial I/O (in clock-synchronous serial I/O mode)

Transmission/reception

(1) With an external clock selected, and choosing the $\overline{RTS}$ function, the output level of the $\overline{RTSi}$ pin goes to “L” when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the $\overline{RTSi}$ pin goes to “H” when reception starts. So if the $\overline{RTSi}$ pin is connected to the $\overline{CTSi}$ pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the $\overline{RTS}$ function has no effect. Figure 2.3.11 shows an example of wiring.

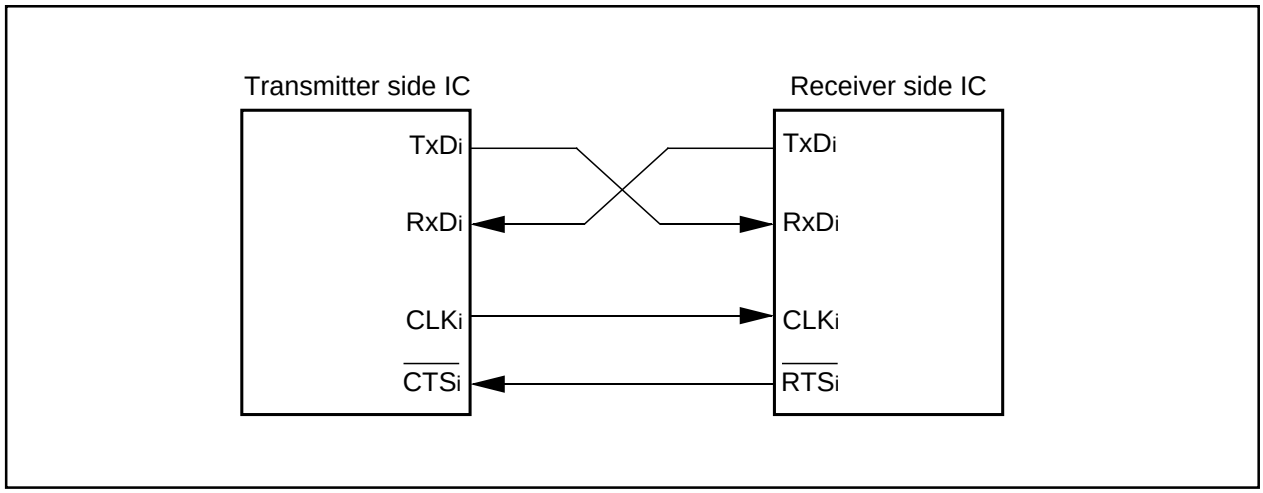


Figure 2.3.11. Example of wiring

Transmission

(1) With an external clock selected, perform the following set-up procedure with the CLKi pin input level = “H” if the CLK polarity select bit = “0” or with the CLKi pin input level = “L” if the CLK polarity select bit = “1”:
   1. Set the transmit enable bit (to “1”)
   2. Write transmission data to the UARTi transmit buffer register
   3. “L” level input to the $\overline{\text{CTSi}}$ pin (when the $\overline{\text{CTS}}$ function is selected)

Reception (1) In operating the clock-synchronous serial I/O, operating a transmitter generates a shift clock. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TxDi pin (transmission pin) when receiving data.

(2) With the internal clock selected, setting the transmit enable bit to “1” (transmission-enabled status) and setting dummy data in the UARTi transmission buffer register generates a shift clock.
With the external clock selected, a shift clock is generated when the transmit enable bit is set to “1”, dummy data is set in the UARTi transmit buffer register, and the external clock is input to the CLKi pin.

(3) When receiving data in succession, an overrun error occurs if the serial interface starts receiving the next data item while the receive complete flag is 1 (before reading the contents of the UARTi receive buffer register) and receives the 7th bit of the next data item, and then the overrun error flag is set to “1”. In this instance, the next data is written to the UARTi receive buffer register, so handle with this problem by writing programs on transmission side and reception side so that the previous data is transmitted again.
If an overrun error occurs, the UARTi receive interrupt request bit does not go to “1”.

(4) To receive data in succession, set dummy data in the lower-order byte of the UARTi transmit buffer register every time reception is made. In continuous receive mode, when the receive buffer is read out,the unit simultaneously goes to a receive enable state without having to set dummy data back to the transmit buffer register again.

(5) With an external clock selected, perform the following set-up procedure with the CLKi pin input level = “H” if the CLK polarity select bit = “0” or with the CLKi pin input level = “L” if the CLK polarity select bit = “1”:
   1. Set receive enable bit (to “1”)
   2. Set transmit enable bit (to “1”)
   3. Write dummy data to the UARTi transmit buffer register

(6) Output from the $\overline{\text{RTS}}$ pin goes to “L” level as soon as the receive enable bit is set to “1”. This is not related to the content of the transmit buffer empty flag or the content of the transmit enable bit. Output from the RTS pin goes to “H” level when reception starts, and goes to “L” level when reception is completed. This is not related to the content of the transmit buffer empty flag or the content of the receive complete flag.

2.4 Clock-Asynchronous Serial I/O (UART)

2.4.1 Overview

UART handles communications by means of character-by-character synchronization. The transmission side and the reception side are independent of each other, so full-duplex communication is possible. The following is an overview of the clock-asynchronous serial I/O.

(1) Transmission/reception format

Figure 2.4.1 shows the transmission/reception format, and Table 2.4.1 shows the names and functions of transmission data.

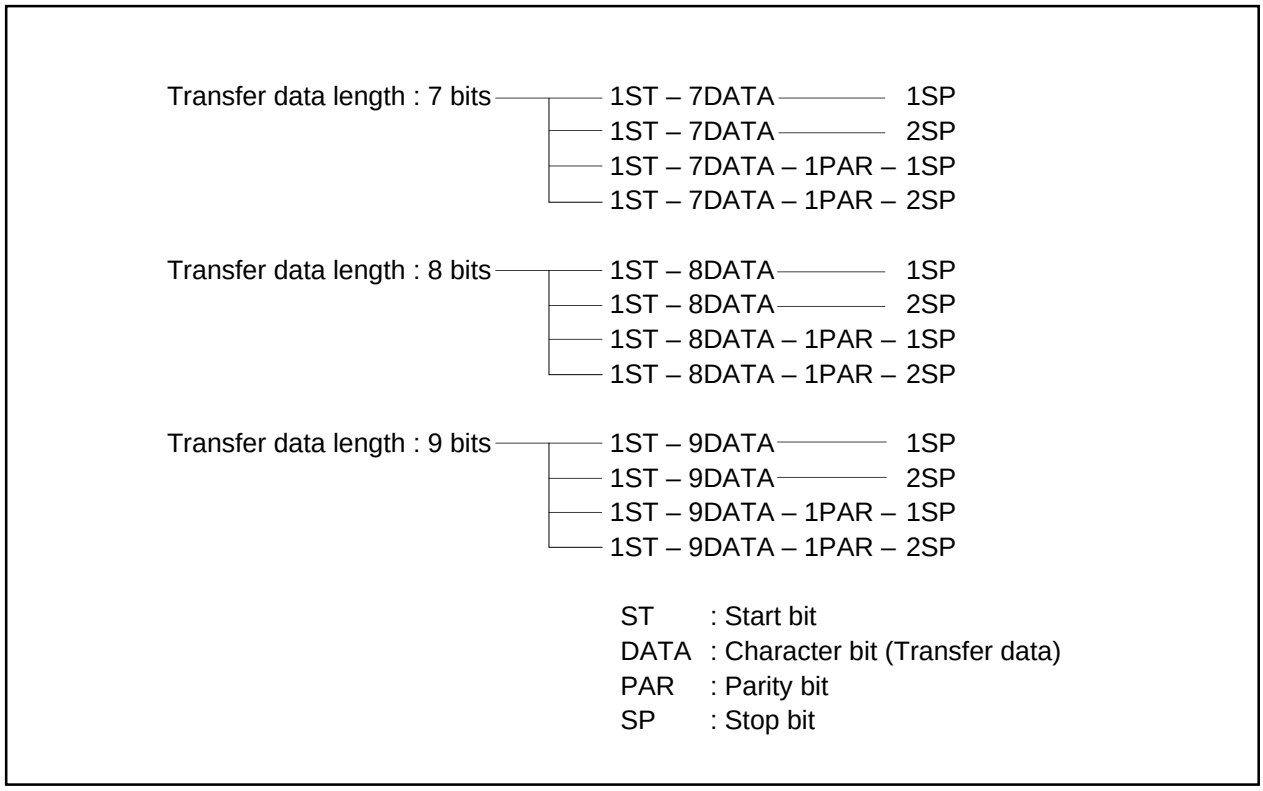

Transfer data length : 7 bits
1ST – 7DATA – 1SP
1ST – 7DATA – 2SP
1ST – 7DATA – 1PAR – 1SP
1ST – 7DATA – 1PAR – 2SP
Transfer data length : 8 bits
1ST – 8DATA – 1SP
1ST – 8DATA – 2SP
1ST – 8DATA – 1PAR – 1SP
1ST – 8DATA – 1PAR – 2SP
Transfer data length : 9 bits
1ST – 9DATA – 1SP
1ST – 9DATA – 2SP
1ST – 9DATA – 1PAR – 1SP
1ST – 9DATA – 1PAR – 2SP
ST : Start bit
DATA : Character bit (Transfer data)
PAR : Parity bit
SP : Stop bit

Figure 2.4.1. Transmission/reception format

Table 2.4.1. Transmission data names and functions

Name	Function
ST (start bit)	A 1-bit “L” signal to be added immediately before character bits. This bit signals the start of data transmission.
DATA (character bits)	Transmission data set in the UARTi transmit buffer register.
PAR (parity bit)	A signal to be added immediately after character bits so as to increase data reliability. The level of this signal so varies that the total number of 1's in character bits and this bit always becomes even or odd depending on which parity is chosen, even or odd.
SP (stop bit)	Either 1-bit or 2-bit “H” signal to be added immediately after character bits (after the parity bit if parity is checked). This / they signals the end of data transmission.

(2) Transfer rate

The divide-by-16 frequency, resulting from division in the bit rate generator (BRG), becomes the transfer rate. The count source for the transfer rate register can be selected from f_1, f_8, f_{32}, and the input from the CLK pin. Clocks f_1, f_8, f_{32} are derived by dividing the CPU's main clock by 1, 8, and 32 respectively.

Table 2.4.2. Example of baud rate setting

Baud rate (bps)	BRG's count source	System clock : 16MHz		System clock : 7.3728MHz	
		BRG's set value : n	Actual time (bps)	BRG's set value : n	Actual time (bps)
600	f_8	207 (CF_{16})	601	95 ($5F_{16}$)	600
1200	f_8	103 (67_{16})	1202	47 ($2F_{16}$)	1200
2400	f_8	51 (33_{16})	2404	23 (17_{16})	2400
4800	f_1	207 (CF_{16})	4808	95 ($5F_{16}$)	4800
9600	f_1	103 (67_{16})	9615	47 ($2F_{16}$)	9600
14400	f_1	68 (44_{16})	14493	31 ($1F_{16}$)	14400
19200	f_1	51 (33_{16})	19231	23 (17_{16})	19200
28800	f_1	34 (22_{16})	28571	15 (F_{16})	28800
31250	f_1	33 (21_{16})	31250		

(3) An error detection

In UART mode, detected errors are shown in Table 2.4.3.

Table 2.4.3. Error detection

Type of error	Description	When the flag turns on	How to clear the flag
Overrun error	• This error occurs when the serial interface starts receiving the next data item before reading the contents of the UARTi receive buffer register and receives the bit preceding the final stop bit of the next data item. • The contents of the UARTi receive buffer register are undefined. • The UARTi receive interrupt request bit does not go to "1".	The error is detected when data is transferred from the UARTi receive register to the UARTi receive buffer register.	• Set the serial I/O mode select bits to "000_2". • Set the receive enable bit to "0".
Framing error	• This error occurs when the stop bit falls short of the set number of stop bits.		• Set the serial I/O mode select bits to "000_2". • Set the receive enable bit to "0". • Read the lower-order byte of the UARTi receive buffer register.
Parity error	• With parity enabled, this error occurs when the total number of 1's in character bits and the parity bit is different from the specified number.		
Error-sum flag	• This flag turns on when any error (overrun, framing, or parity) is detected.		• When all error (overrun, framing, and parity) are removed, the flag is cleared.

(4) How to deal with an error

When receiving data, read an error flag and reception data simultaneously to determine which error has occurred. If the data read is erroneous, initialize the error flag and the UARTi receive buffer register, then receive the data again.

To initialize the UARTi receive buffer register

1. Set the receive enable bit to “0” (disable reception).
2. Set the receive enable bit to “1” again (enable reception).

To transmit data again due to an error on the reception side, set the UARTi transmit buffer register again, then transmit the data again.

To set the UARTi transmit buffer register again

1. Set the serial I/O mode select bits to “000_2” (invalidate serial I/O).
2. Set the serial I/O mode select bits again.
3. Set the transmit enable bit to “1” (enable transmission), then set transmission data in the UARTi transmit buffer register.

(5) Functions selection

In operating UART, the following functions can be used:

(a) $\overline{\text{CTS}}/\overline{\text{RTS}}$ function

$\overline{\text{CTS}}$ function is a function in which an external IC can start transmission/reception by means of inputting an “L” level to the $\overline{\text{CTS}}$ pin. The $\overline{\text{CTS}}$ pin input level is detected when transmission/reception starts, so if the level is gone to“ H” while transmission/reception is in progress, transmission/reception stops at the next data.

$\overline{\text{RTS}}$ function is a function to inform an external IC that $\overline{\text{RTS}}$ pin output level has changed to “L” when reception is ready. $\overline{\text{RTS}}$ regoes to “H” at the first falling edge of the transfer clock.

When using clock-asynchronous serial I/O, choose one of three types of $\overline{\text{CTS}}/\overline{\text{RTS}}$ functions.

- $\overline{\text{CTS}}/\overline{\text{RTS}}$ functions disabled — $\overline{\text{CTS}}/\overline{\text{RTS}}$ pin is a programmable I/O port.
- $\overline{\text{CTS}}$ function only enabled — $\overline{\text{CTS}}/\overline{\text{RTS}}$ pin performs the $\overline{\text{CTS}}$ function.
- $\overline{\text{RTS}}$ function only enabled — $\overline{\text{CTS}}/\overline{\text{RTS}}$ pin performs the $\overline{\text{RTS}}$ function.

(b) Data logic select function

This function is to reverse data when writing to transmit buffer register or reading from receive buffer register.

(c) LSB/MSB first select function

This function is to choose whether to transmit/receive data from bit 0 or bit 7. This is valid when the transfer data length is 8 bits long.

Choose either of the following:

- LSB first Data is transmitted/received from bit 0.
- MSB first Data is transmitted/received from bit 7.

(d) TxD, RxD I/O polarity reverse function

This function is to reverse a polarity of TxD port output level and a polarity of RxD port input level.

(e) Bus collision detection function

This function is to sample the output level of the TxD pin and the input level of the RxD pin; if their values are different, then an interrupt request occurs.

The following examples are described in section 2.4.2 and 2.4.3.

- Transmission WITH: $\overline{\text{CTS}}$ function, WITHOUT: other functions
- Reception WITH: $\overline{\text{RTS}}$ function, WITHOUT: other functions

Also, the SIM interface is used by adding some extra settings in clock-asynchronous serial I/O mode. Direct or inverse format is selected by connecting SIM card.

The following examples are described in section 2.4.4 and 2.4.5.

- Transmission WITH: direct format
- Reception WITH: direct format

(6) Input to the serial I/O and the direction register

To input an external signal to the serial I/O, set the direction register of the relevant port to input.

(7) Pins related to the serial I/O

- $\overline{\text{CTS}}_0$, $\overline{\text{CTS}}_1$, $\overline{\text{CTS}}_2$, $\overline{\text{CTS}}_3$ pins :Input pins for the $\overline{\text{CTS}}$ function
- $\overline{\text{RTS}}_0$, $\overline{\text{RTS}}_1$, $\overline{\text{RTS}}_2$, $\overline{\text{RTS}}_3$ pins :Output pins for the $\overline{\text{RTS}}$ function
- CLK_0, CLK_1, CLK_2, CLK_3 pins :Input pins for the transfer clock
- RxD_0, RxD_1, RxD_2, RxD_3 pins :Input pins for data
- TxD_0, TxD_1, TxD_2, TxD_3 pins :Output pins for data (Note)

Note: Since TxD_2 pin is N-channel open drain, this pin needs pull-up resistor.

(8) Registers related to the serial I/O

Figure 2.4.2 shows the memory map of serial I/O-related registers, and Figures 2.4.3 to 2.4.6 show UARTi-related registers.

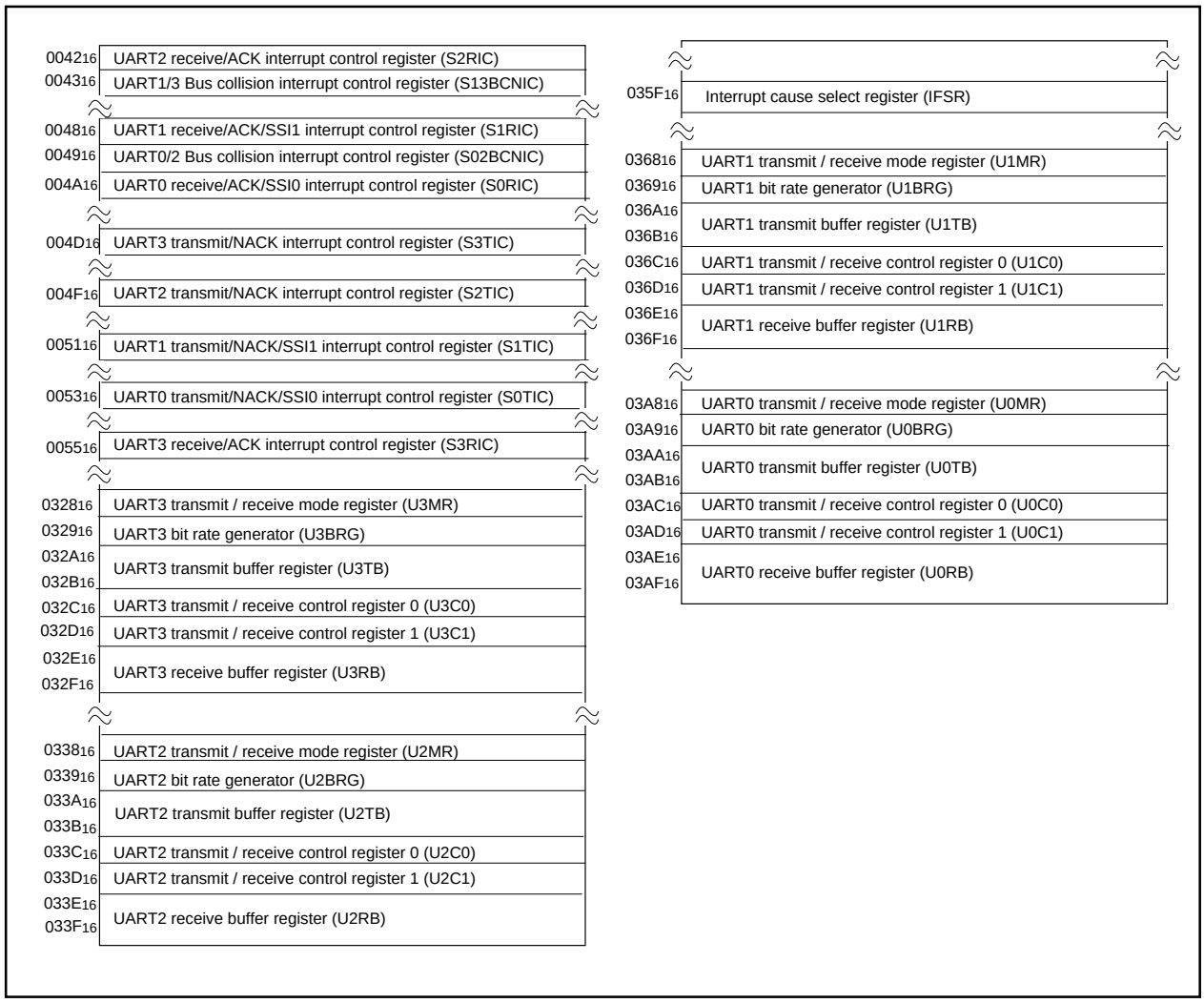
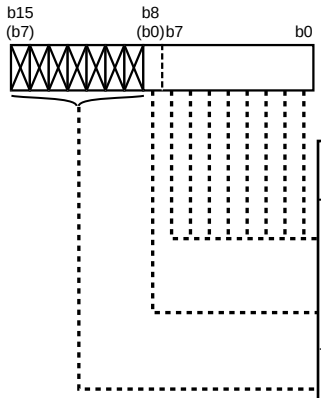

0042 UART2 receive/ACK interrupt control register (S2RIC)
0043 UART1/3 Bus collision interrupt control register (S13BCNIC)
0048 UART1 receive/ACK/SSI1 interrupt control register (S1RIC)
0049 UART0/2 Bus collision interrupt control register (S02BCNIC)
004A UART0 receive/ACK/SSI0 interrupt control register (S0RIC)
004D UART3 transmit/NACK interrupt control register (S3TIC)
004F UART2 transmit/NACK interrupt control register (S2TIC)
0051 UART1 transmit/NACK/SSI1 interrupt control register (S1TIC)
0053 UART0 transmit/NACK/SSI0 interrupt control register (S0TIC)
0055 UART3 receive/ACK interrupt control register (S3RIC)
0328 UART3 transmit / receive mode register (U3MR)
0329 UART3 bit rate generator (U3BRG)
032A 032B UART3 transmit buffer register (U3TB)
032C UART3 transmit / receive control register 0 (U3C0)
032D UART3 transmit / receive control register 1 (U3C1)
032E 032F UART3 receive buffer register (U3RB)
0338 UART2 transmit / receive mode register (U2MR)
0339 UART2 bit rate generator (U2BRG)
033A 033B UART2 transmit buffer register (U2TB)
033C UART2 transmit / receive control register 0 (U2C0)
033D UART2 transmit / receive control register 1 (U2C1)
033E 033F UART2 receive buffer register (U2RB)
035F Interrupt cause select register (IFSR)
0368 UART1 transmit / receive mode register (U1MR)
0369 UART1 bit rate generator (U1BRG)
036A 036B UART1 transmit buffer register (U1TB)
036C UART1 transmit / receive control register 0 (U1C0)
036D UART1 transmit / receive control register 1 (U1C1)
036E 036F UART1 receive buffer register (U1RB)
03A8 UART0 transmit / receive mode register (U0MR)
03A9 UART0 bit rate generator (U0BRG)
03AA 03AB UART0 transmit buffer register (U0TB)
03AC UART0 transmit / receive control register 0 (U0C0)
03AD UART0 transmit / receive control register 1 (U0C1)
03AE 03AF UART0 receive buffer register (U0RB)

Figure 2.4.2. Memory map of UARTi-related registers

UARTi transmit buffer register (i= 0 to 3) (Note)

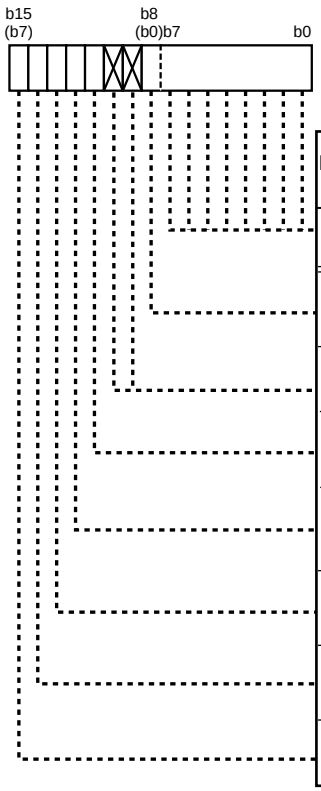


Symbol	Address	When reset
U0TB	$03AB_{16}$, $03AA_{16}$	Indeterminate
U1TB	$036B_{16}$, $036A_{16}$	Indeterminate
U2TB	$033B_{16}$, $033A_{16}$	Indeterminate
U3TB	$032B_{16}$, $032A_{16}$	Indeterminate

Bit Symbol	Function (clock synchronous serial I/O mode)	Function (UART mode)	R	W
———	Transmit data	Transmit data	×	○
———	———	Transmit data (9th bit)	×	○
———	Nothing is assigned. Write “0” when writing to these bits. The values are indeterminate when read.		–	–

Note: Use MOV instruction to write to this register.

UARTi receive buffer register (i= 0 to 3)



Symbol	Address	When reset
U0RB	$03AF_{16}$, $03AE_{16}$	Indeterminate
U1RB	$036F_{16}$, $036E_{16}$	Indeterminate
U2RB	$033F_{16}$, $033E_{16}$	Indeterminate
U3RB	$032F_{16}$, $032E_{16}$	Indeterminate

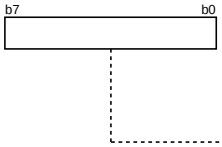
Bit Symbol	Bit Name	Function (clock synchronous serial I/O mode)	Function (UART mode)	R	W
———	———	Receive data	Receive data	○	×
———	———	———	Receive data (9th bit)	○	×
———	Nothing is assigned. Write “0” when writing to these bits. The values are indeterminate when read.			–	–
ABT	Arbitration lost detecting flag (Note 1)	0 : Not detected 1 : Detected	Invalid	○	○
OER	Overrun error flag (Note 2)	0 : No overrun error 1 : Overrun error	0 : No overrun error 1 : Overrun error	○	×
FER	Framing error flag (Note 2)	Invalid	0 : No framing error 1 : Framing error	○	×
PER	Parity error flag (Note 2)	Invalid	0 : No parity error 1 : Parity error	○	×
SUM	Error sum flag (Note 2)	Invalid	0 : No error 1 : Error	○	×

Note 1: Always write “0”.

Note 2: Bits 15 to 12 are set to “0000_2” when the serial I/O mode select bit (bits 0 to 2 at addresses $03A8_{16}$, 0368_{16}, 0338_{16}, 0328_{16}) are set to “000_2” or the receive enable bit is set to “0”. Bit 15 is set to “0” when all of bits 14 to 12 are set to “0”.
Bits 14 and 13 are also set to “0” when the lower byte of the UARTi receive buffer register (addresses $03AE_{16}$, $036E_{16}$, $033E_{16}$, $032E_{16}$) is read.

Figure 2.4.3. UARTi-related registers (1)

UARTi bit rate generator (o=0 to 3) (Note 1, 2)

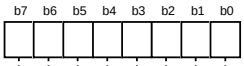


Symbol	Address	When reset
U0BRG	$03A9_{16}$	Indeterminate
U1BRG	0369_{16}	Indeterminate
U2BRG	0339_{16}	Indeterminate
U3BRG	0329_{16}	Indeterminate

Function	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}	—	○

Note 1: Use MOV instruction to write to this register.
Note 2: Write a value to this register while transmit/receive halts.

UARTi transmit/receive mode register (i = 0 to 3)



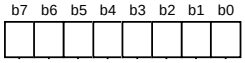
Symbol	Address	When reset
U0MR	$03A8_{16}$	00_{16}
U1MR	0368_{16}	00_{16}
U2MR	0338_{16}	00_{16}
U3MR	0328_{16}	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit (Note 3)	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Serial I/O mode 0 1 1 : I^2C mode Inhibited except in cases listed above	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long Inhibited except in cases listed above	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 1)	0 : Internal clock 1 : External clock (Note 1)	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	TxD, RxD input/output polarity switch bit (Note 2)	0 : Normal 1 : Reversed		○	○

Note 1: When I^2C bus interface mode is selected, set the port direction register for the corresponding port (SCLi) to 0, or the port direction register to 1 and the port data register to 1. When a mode other than serial I/O mode is selected, set the port direction register for the corresponding port (CLKi) to 0.
Note 2: Normally set to “0”.
Note 3: Set the RxDi pin's port direction register to “0” when receiving.

Figure 2.4.4. UARTi-related registers (2)

UARTi transmit/receive control register 0 (i=0 to 3)

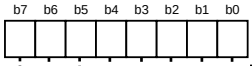


Symbol: UiC0 (i=0 to 3) | Address: 03AC16, 036C16, 033C16, 032C16, | When reset: 0816

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f1 is selected 0 1 : f8 is selected 1 0 : f32 is selected 1 1 : Inhibited	b1 b0 **0 0 : f1 is selected** **0 1 : f8 is selected** **1 0 : f32 is selected** **1 1 : Inhibited**	○	○
CLK1				○	○
CRS	CTS/RTS function select bit	Valid when bit 4 = "0" 0 : CTS function is selected (Note 1) 1 : RTS function is selected (Note 4)	Valid when bit 4 = "0" 0 : CTS function is selected (Note 1) 1 : RTS function is selected (Note 4)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	CTS/RTS disable bit	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled	○	○
NCH (Note 2)	Data output select bit	0 : TxDi/SDAi and SCLi pin is CMOS output 1 : TxDi/SDAi and SCLi pin is N-channel open drain output	0 : TxDi/SDAi and SCLi pin is CMOS output 1 : TxDi/SDAi and SCLi pin is N-channel open drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Set to "0"	○	○
UFORM	Transfer format select bit (Note 3)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

Note 1: Set the corresponding port direction register to "0".
Note 2: UART2 transfer pin (TxD2: P70 and SCL2: P71) is N-channel open drain output. It cannot be set to CMOS output.
Note 3: Only clock synchronous serial I/O mode and 8-bit UART mode are valid.
Note 4: The corresponding port register and port direction register are invalid.

UARTi transmit/receive control register 1 (i= 0 to 3)



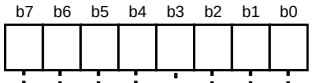
Symbol: UiC1 (i=0 to 3) | Address: 03AD16, 036D16, 033D16, 032D16 | When reset: 0216

Bit Symbol	Bit Name	Function (clock synchronous serial I/O mode)	Function (UART mode)	R	W
TE	Transmit enable bit	0 : Transmit disabled 1 : Transmit enabled		○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register		○	○
RE	Receive enable bit	0 : Receive disabled 1 : Receive enabled		○	○
RI	Receive complete flag	0 : Data packet in receive buffer register 1 : No data packet in receive buffer register		○	○
UiIRS	UARTi transmit interrupt cause select bit	0 : Transmit buffer empty (TI =1) 1 : Transmit buffer completed (TXEPT =1)		○	○
UiRRM	UARTi continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Set to "0"	○	○
UiLCH	Data logic select bit	0 : No reverse 1 : Reverse		○	○
UiERE	Error signal output enable bit	Set to "0" The value is indeterminate when read.	0 : Output disabled 1 : Output enabled (Note 1)	○	○

Note 1: When disabling the error signal output, set the UiERE bit to "0" after setting the UiMR register.

Figure 2.4.5. UARTi-related registers (3)

Interrupt request cause select register



Symbol IFSR | Address 035F16, | When reset 0016

Bit Symbol	Bit name	Function	R	W
IFSR0	INT0 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
IFSR1	INT1 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
IFSR2	INT2 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
Nothing is assigned. Write “0” when writing to this bit. The value is indeterminate when read.			—	—
IFSR6	Bus collision interrupt request cause select bit 0	0 : UART0 Bus collision / Start/stop condition detection / Trouble error detection 1 : UART2 Bus collision / Start/stop condition detection / Trouble error detection	○	○
IFSR7	Bus collision interrupt request cause select bit 1	0 : UART1 Bus collision / Start/stop condition detection / Trouble error detection 1 : UART3 Bus collision / Start/stop condition detection / Trouble error detection	○	○

Figure 2.4.6. UARTi-related registers (4)

2.4.2 Operation of Serial I/O (transmission in UART mode)

In transmitting data in UART mode, choose functions from those listed in Table 2.4.4. Operations of the circled items are described below. Figure 2.4.7 shows the operation timing, and Figures 2.4.8 and 2.4.9 show the set-up procedures.

Table 2.4.4. Choosed functions

Item	Set-up		Item	Set-up	
Transfer clock source	O	Internal clock (f_1 / f_8 / f_{32})	Data logic select function	O	No reverse
		External clock (CLKi pin)			Reverse
$\overline{\text{CTS}}$ function	O	$\overline{\text{CTS}}$ function enabled	TxD, RxD I/O polarity reverse bit	O	No reverse
		$\overline{\text{CTS}}$ function disabled			Reverse
Transmission interrupt factor		Transmission buffer empty	Bus collision detection function	O	Not selected
	O	Transmission complete			Selected

Operation

(1) Setting the transmit enable bit to “1” and writing transmission data to the UARTi transmit buffer register readies the data transmissible status.

(2) When input to the $\overline{\text{CTSi}}$ pin goes to “L”, transmission starts (the $\overline{\text{CTSi}}$ pin needs to be controlled on the reception side).

(3) Transmission data held in the UARTi transmit buffer register is transmitted to the UARTi transmit register. At this time, the first bit (the start bit) of the transmission data is transmitted from the TxDi pin. Then, data is transmitted, bit by bit, in sequence: LSB, ···, MSB, parity bit, and stop bit(s).

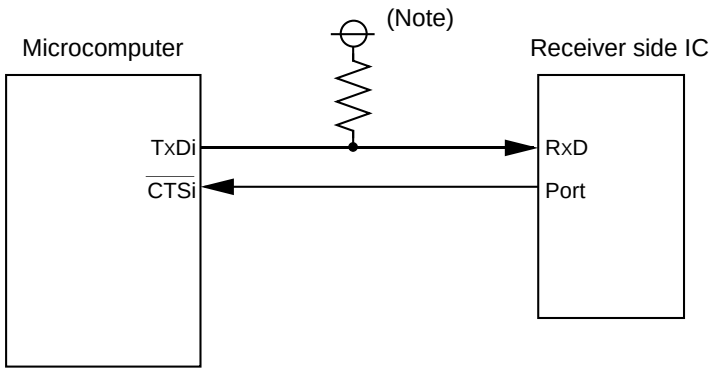
(4) When the stop bit(s) is (are) transmitted, the transmit register empty flag goes to “1”, which indicates that transmission is completed. At this time, the UARTi transmit interrupt request bit goes to “1”. The transfer clock stops at “H” level.

(5) If the transmission condition of the next data is ready when transmission is completed, a start bit is generated following to stop bit(s), and the next data is transmitted.

Note

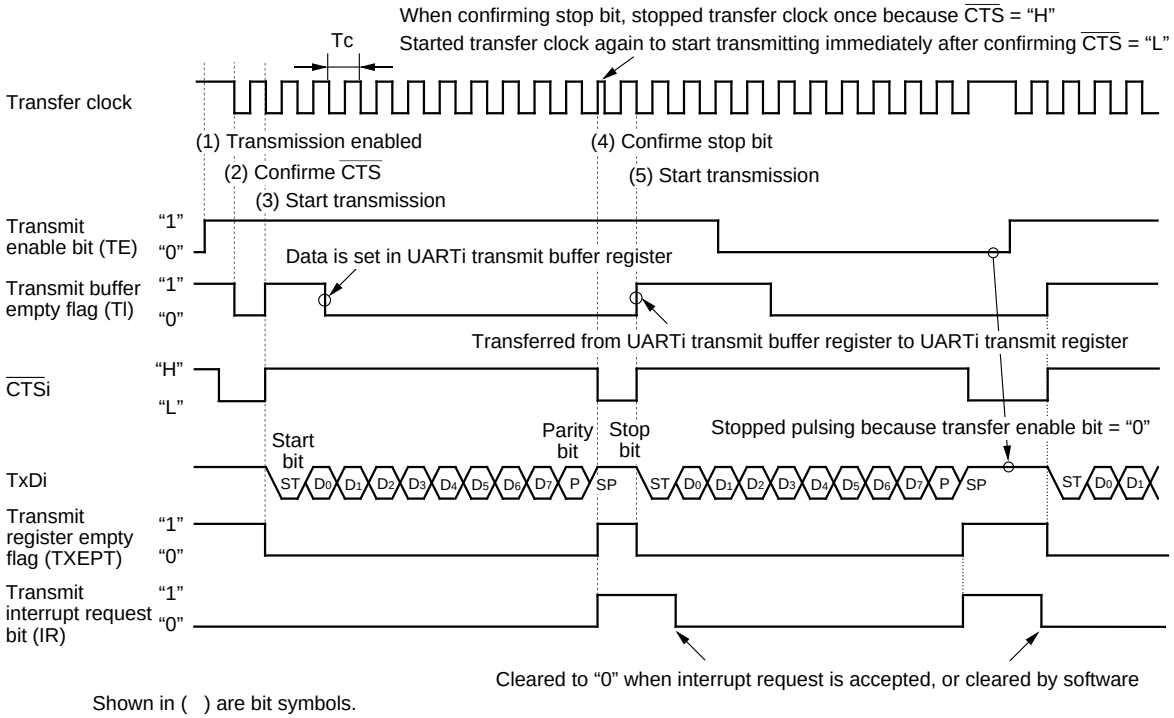
• Set $\overline{\text{CTSi}}$ pin's port direction register to “0”.

Example of wiring



Note: Since TxD2 pin is N-channel open drain, this pin needs pull-up resistance.

Example of operation



Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- CTS function is selected.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 2.4.7. Operation timing of transmission in UART mode

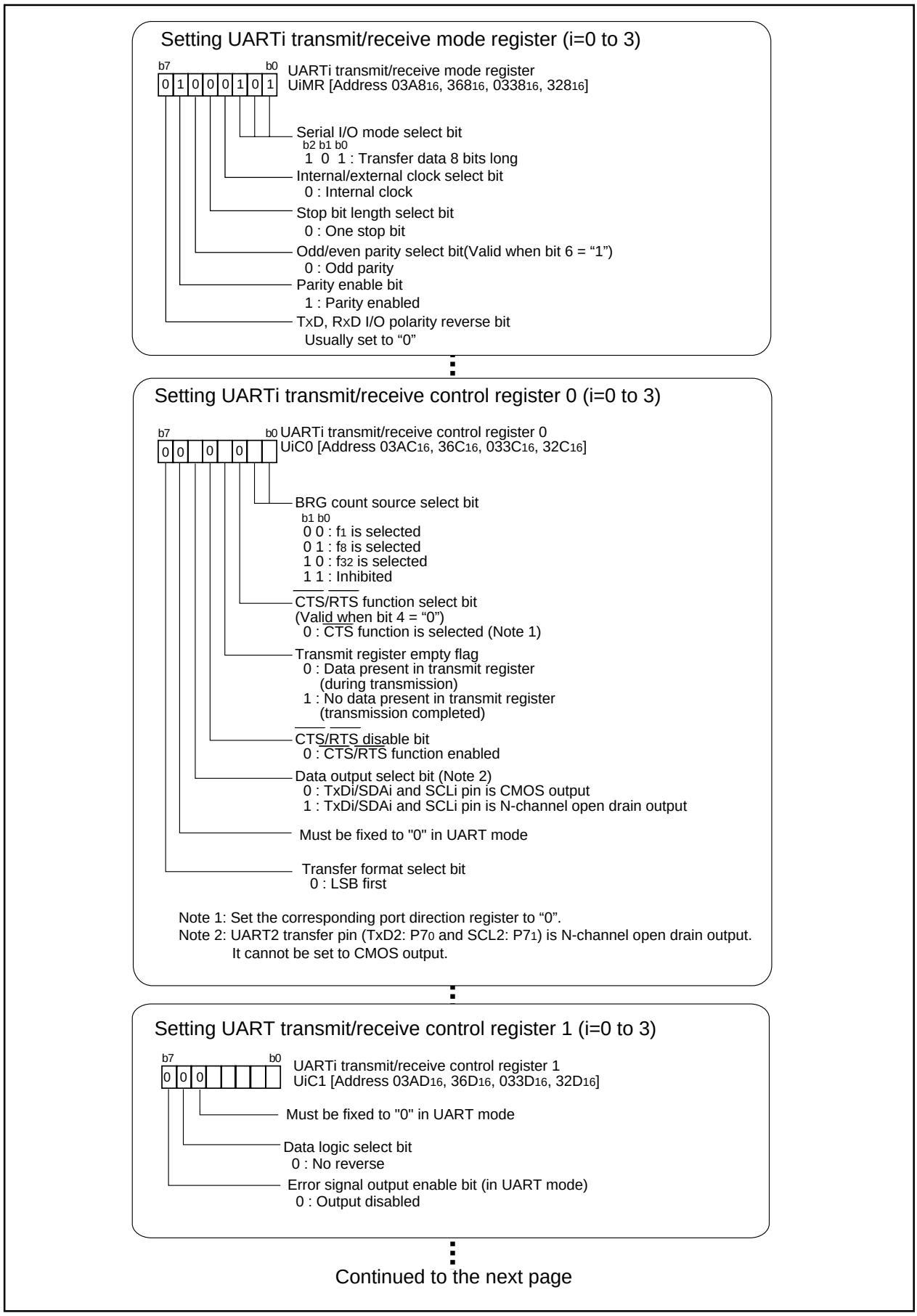

Setting UARTi transmit/receive mode register (i=0 to 3)
b7 b0
0 1 0 0 0 1 0 1
UARTi transmit/receive mode register
UiMR [Address 03A816, 36816, 033816, 32816]
Serial I/O mode select bit
b2 b1 b0
1 0 1 : Transfer data 8 bits long
Internal/external clock select bit
0 : Internal clock
Stop bit length select bit
0 : One stop bit
Odd/even parity select bit(Valid when bit 6 = "1")
0 : Odd parity
Parity enable bit
1 : Parity enabled
TxD, RxD I/O polarity reverse bit
Usually set to "0"
Setting UARTi transmit/receive control register 0 (i=0 to 3)
b7 b0
0 0 0 0
UARTi transmit/receive control register 0
UiC0 [Address 03AC16, 36C16, 033C16, 32C16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : Inhibited
CTS/RTS function select bit
(Valid when bit 4 = "0")
0 : CTS function is selected (Note 1)
Transmit register empty flag
0 : Data present in transmit register
(during transmission)
1 : No data present in transmit register
(transmission completed)
CTS/RTS disable bit
0 : CTS/RTS function enabled
Data output select bit (Note 2)
0 : TxDi/SDAi and SCLi pin is CMOS output
1 : TxDi/SDAi and SCLi pin is N-channel open drain output
Must be fixed to "0" in UART mode
Transfer format select bit
0 : LSB first
Note 1: Set the corresponding port direction register to "0".
Note 2: UART2 transfer pin (TxD2: P70 and SCL2: P71) is N-channel open drain output.
It cannot be set to CMOS output.
Setting UART transmit/receive control register 1 (i=0 to 3)
b7 b0
0 0 0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Must be fixed to "0" in UART mode
Data logic select bit
0 : No reverse
Error signal output enable bit (in UART mode)
0 : Output disabled
Continued to the next page

Figure 2.4.8. Set-up procedure of transmission in UART mode (1)

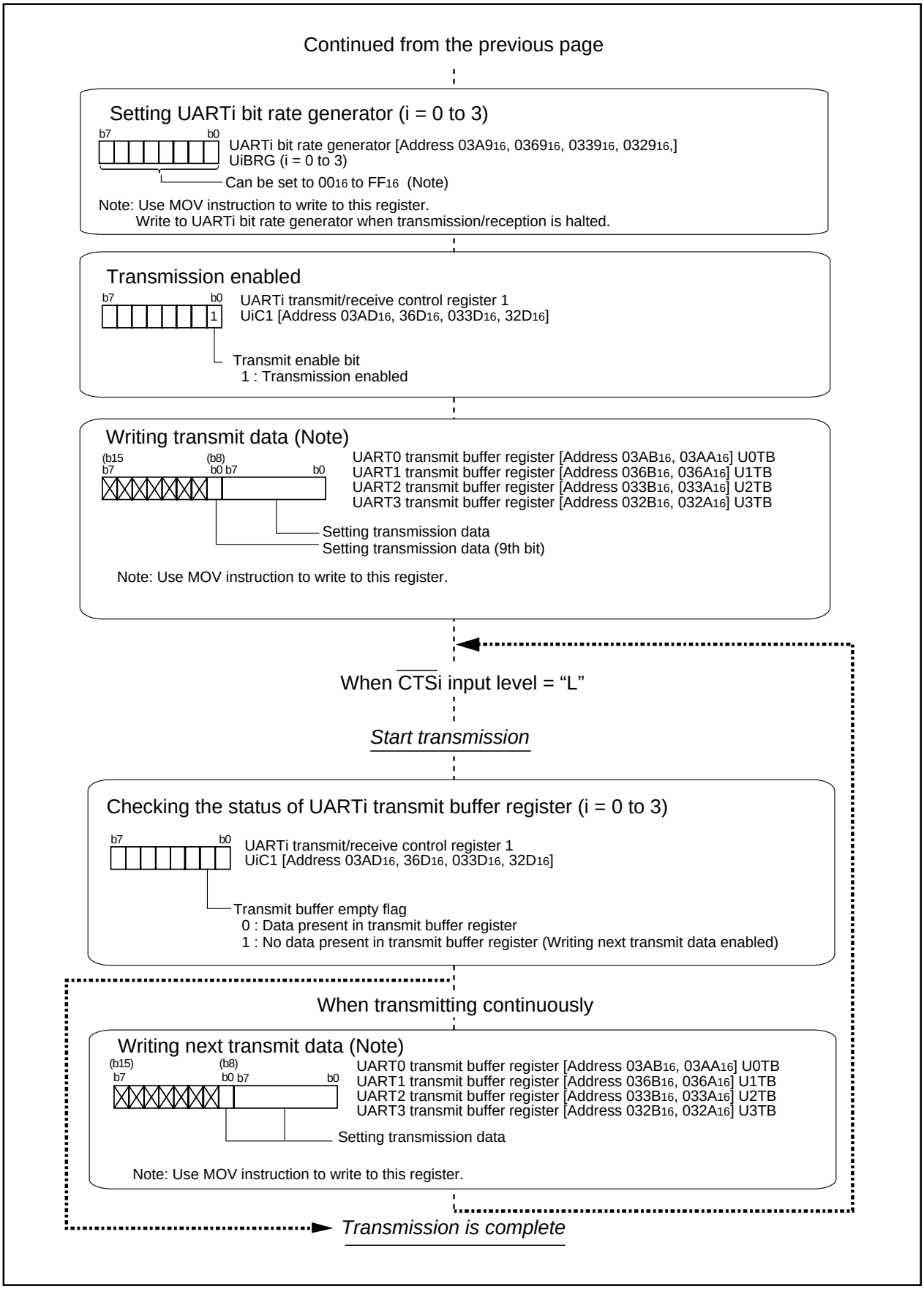

Continued from the previous page
Setting UARTi bit rate generator (i = 0 to 3)
b7 b0
UARTi bit rate generator [Address 03A916, 036916, 033916, 032916,]
UiBRG (i = 0 to 3)
Can be set to 0016 to FF16 (Note)
Note: Use MOV instruction to write to this register.
Write to UARTi bit rate generator when transmission/reception is halted.
Transmission enabled
b7 b0
1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit enable bit
1 : Transmission enabled
Writing transmit data (Note)
(b15) (b8)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting transmission data
Setting transmission data (9th bit)
Note: Use MOV instruction to write to this register.
When CTSi input level = “L”
Start transmission
Checking the status of UARTi transmit buffer register (i = 0 to 3)
b7 b0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit buffer empty flag
0 : Data present in transmit buffer register
1 : No data present in transmit buffer register (Writing next transmit data enabled)
When transmitting continuously
Writing next transmit data (Note)
(b15) (b8)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting transmission data
Note: Use MOV instruction to write to this register.
Transmission is complete

Figure 2.4.9. Set-up procedure of transmission in UART mode (2)

2.4.3 Operation of Serial I/O (reception in UART mode)

In receiving data in UART mode, choose functions from those listed in Table 2.4.5. Operations of the circled items are described below. Figure 2.4.10 shows the operation timing, and Figures 2.4.11 and 2.4.12 show the set-up procedures.

Table 2.4.5. Choosed functions

Item	Set-up		Item	Set-up	
Transfer clock source	O	Internal clock (f_1 / f_8 / f_{32})	TxD, RxD I/O polarity reverse bit	O	No reverse
		External clock (CLKi pin)			Reverse
$\overline{\text{RTS}}$ function	O	$\overline{\text{RTS}}$ function enabled	Bus collision detection function	O	Not selected
		$\overline{\text{RTS}}$ function disabled			Selected
Data logic select function	O	No reverse			
		Reverse			

Operation

(1) Setting the receive enable bit to “1” readies data-receivable status. At this time, output from the $\overline{\text{RTSi}}$ pin goes to “L” level to inform the transmission side that the receivable status is ready.

(2) When the first bit (the start bit) of reception data is received from the RxDi pin, output from the $\overline{\text{RTS}}$ goes to “H” level. Then, data is received, bit by bit, in sequence: LSB, ⋯, MSB, and stop bit(s).

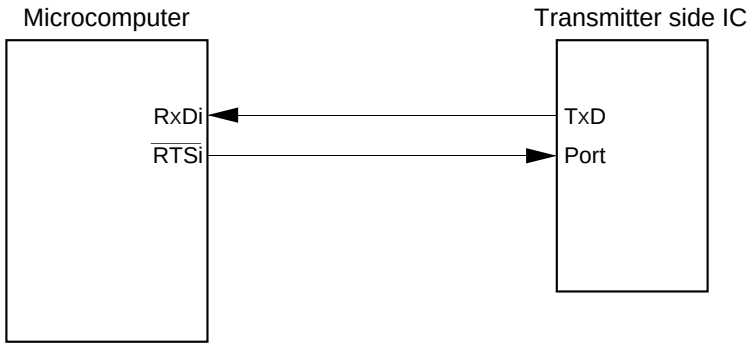
(3) When the stop bit(s) is (are) received, the content of the UARTi receive register is transmitted to the UARTi receive buffer register. At this time, the receive complete flag goes to “1” to indicate that the reception is completed, the UARTi receive interrupt request bit goes to “1”, and output from the $\overline{\text{RTS}}$ pin goes to “L” level.

(4) The receive complete flag goes to “0” when the lower-order byte of the UARTi buffer register is read.

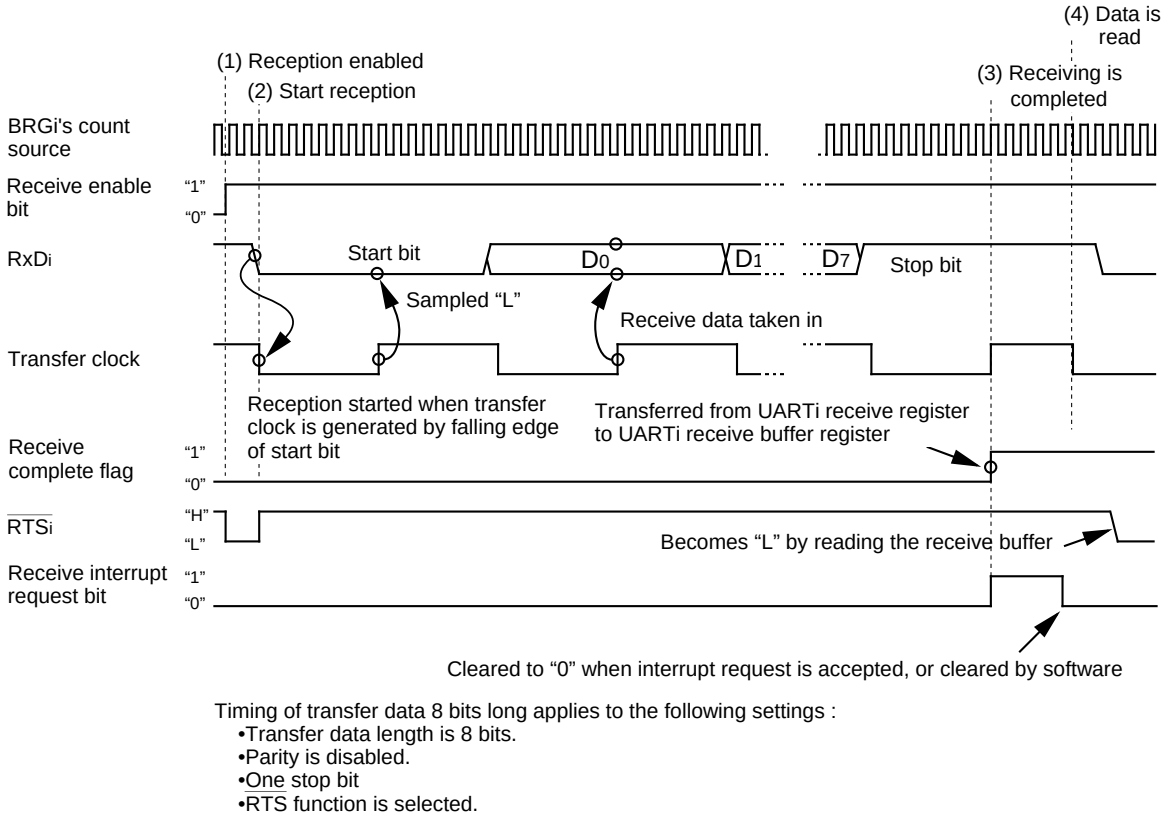
Note

• Set RxDi pin's port direction register to “0”.

Example of wiring



Example of operation



Timing of transfer data 8 bits long applies to the following settings :
- Transfer data length is 8 bits.
- Parity is disabled.
- One stop bit
- RTS function is selected.

Figure 2.4.10. Operation timing of reception in UART mode

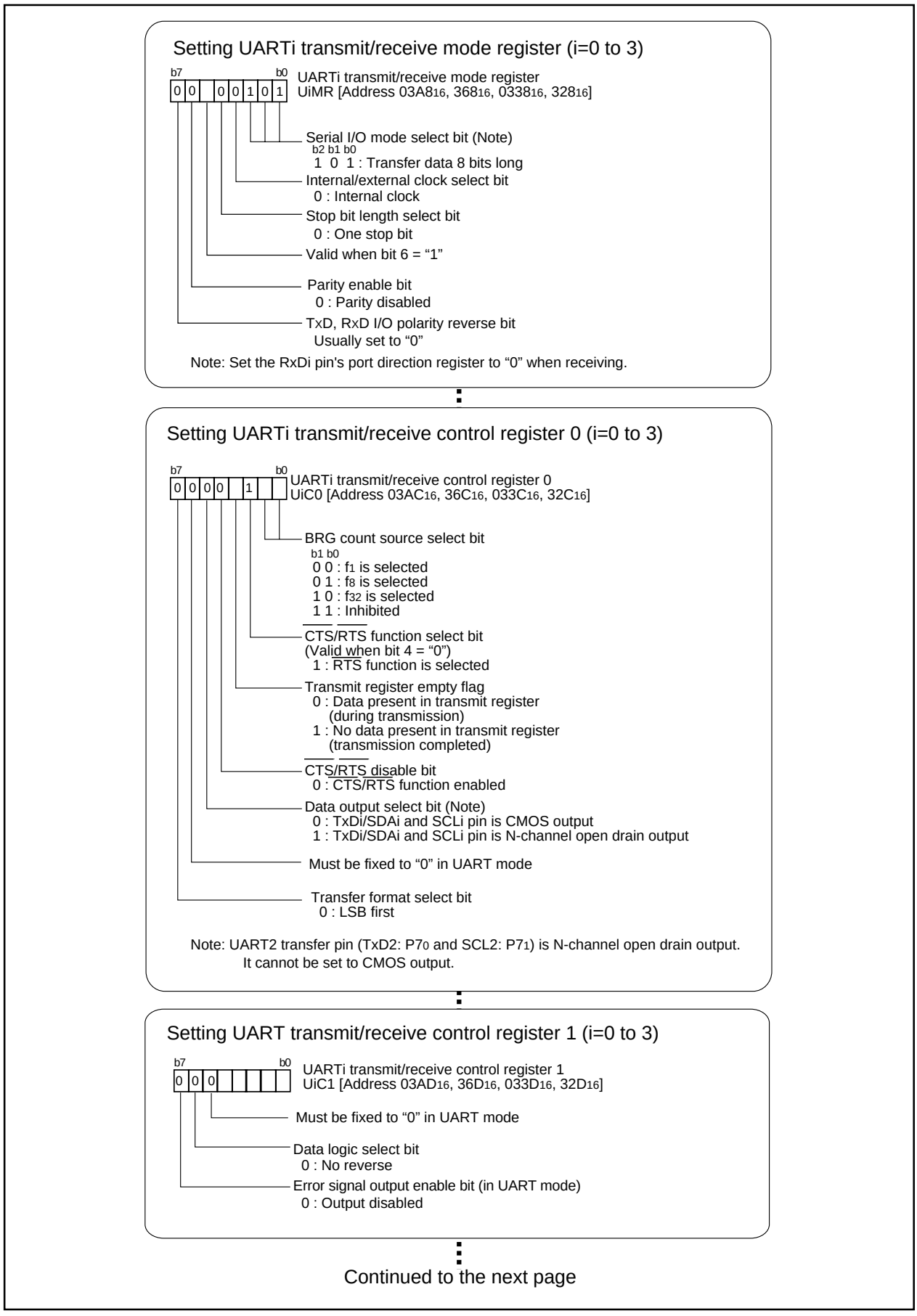

Setting UARTi transmit/receive mode register (i=0 to 3)
b7 b0
0 0 0 0 1 0 1
UARTi transmit/receive mode register
UiMR [Address 03A816, 36816, 033816, 32816]
Serial I/O mode select bit (Note)
b2 b1 b0
1 0 1 : Transfer data 8 bits long
Internal/external clock select bit
0 : Internal clock
Stop bit length select bit
0 : One stop bit
Valid when bit 6 = "1"
Parity enable bit
0 : Parity disabled
TxD, RxD I/O polarity reverse bit
Usually set to "0"
Note: Set the RxDi pin's port direction register to "0" when receiving.
Setting UARTi transmit/receive control register 0 (i=0 to 3)
b7 b0
0 0 0 0 1
UARTi transmit/receive control register 0
UiC0 [Address 03AC16, 36C16, 033C16, 32C16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : Inhibited
CTS/RTS function select bit
(Valid when bit 4 = "0")
1 : RTS function is selected
Transmit register empty flag
0 : Data present in transmit register (during transmission)
1 : No data present in transmit register (transmission completed)
CTS/RTS disable bit
0 : CTS/RTS function enabled
Data output select bit (Note)
0 : TxDi/SDAi and SCLi pin is CMOS output
1 : TxDi/SDAi and SCLi pin is N-channel open drain output
Must be fixed to "0" in UART mode
Transfer format select bit
0 : LSB first
Note: UART2 transfer pin (TxD2: P70 and SCL2: P71) is N-channel open drain output.
It cannot be set to CMOS output.
Setting UART transmit/receive control register 1 (i=0 to 3)
b7 b0
0 0 0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Must be fixed to "0" in UART mode
Data logic select bit
0 : No reverse
Error signal output enable bit (in UART mode)
0 : Output disabled
Continued to the next page

Figure 2.4.11. Set-up procedure of reception in UART mode (1)

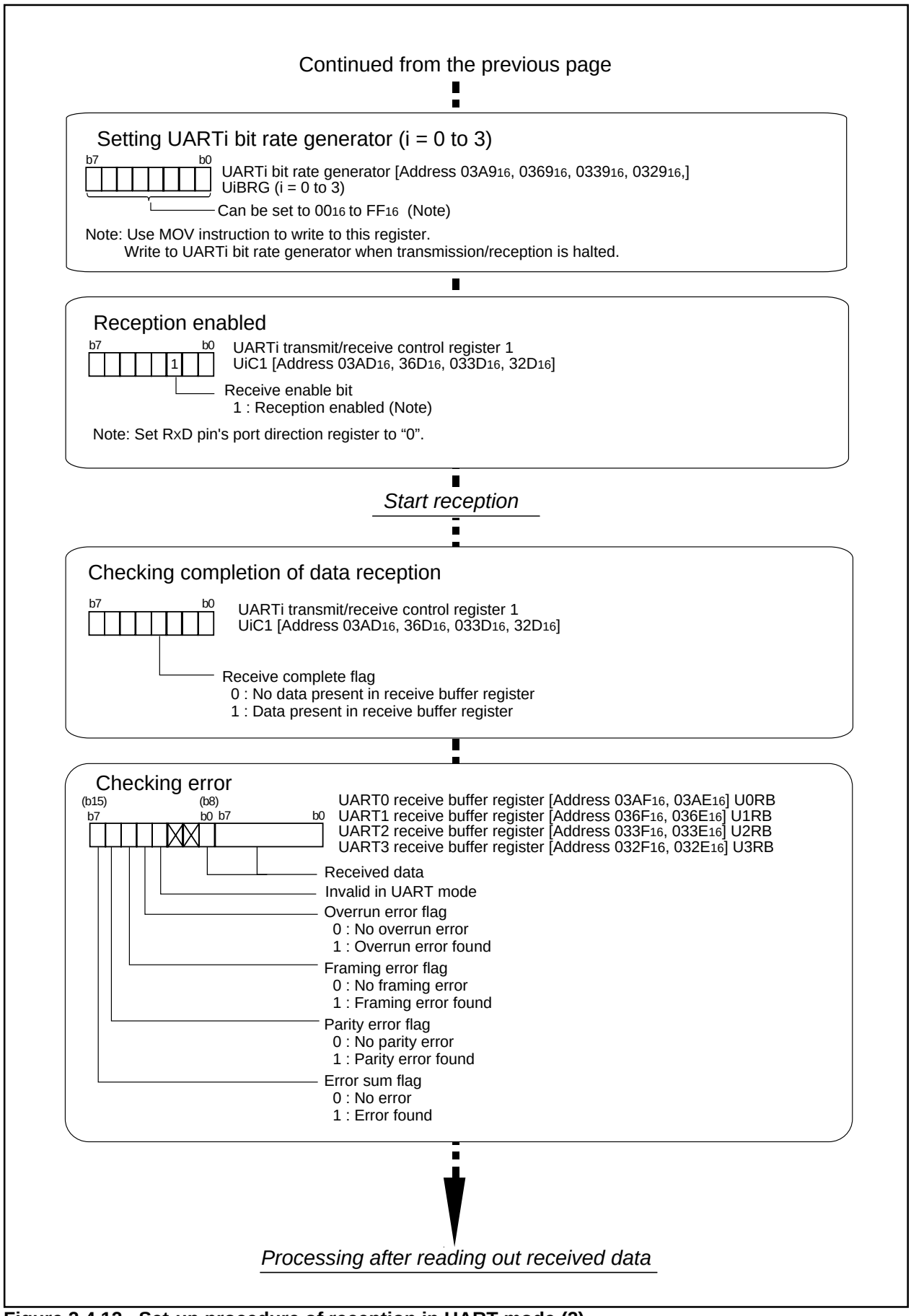

Continued from the previous page
Setting UARTi bit rate generator (i = 0 to 3)
b7
b0
UARTi bit rate generator [Address 03A916, 036916, 033916, 032916,]
UiBRG (i = 0 to 3)
Can be set to 0016 to FF16 (Note)
Note: Use MOV instruction to write to this register.
Write to UARTi bit rate generator when transmission/reception is halted.
Reception enabled
b7
b0
1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Receive enable bit
1 : Reception enabled (Note)
Note: Set RxD pin's port direction register to “0”.
Start reception
Checking completion of data reception
b7
b0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Receive complete flag
0 : No data present in receive buffer register
1 : Data present in receive buffer register
Checking error
(b15)
(b8)
b7
b0 b7
b0
UART0 receive buffer register [Address 03AF16, 03AE16] U0RB
UART1 receive buffer register [Address 036F16, 036E16] U1RB
UART2 receive buffer register [Address 033F16, 033E16] U2RB
UART3 receive buffer register [Address 032F16, 032E16] U3RB
Received data
Invalid in UART mode
Overrun error flag
0 : No overrun error
1 : Overrun error found
Framing error flag
0 : No framing error
1 : Framing error found
Parity error flag
0 : No parity error
1 : Parity error found
Error sum flag
0 : No error
1 : Error found
Processing after reading out received data

Figure 2.4.12. Set-up procedure of reception in UART mode (2)

2.4.4 Serial I/O Precautions (UART Mode)

Description When the level of the CLKi and CTSi pins goes to “H” (Note 1), if the UiMR register is set to any of the following, the UiERE bit in the UiC1 register is set to “1” (parity error signal output enabled). When the PRYE bit in the UiMR register is set to “1” while the UiERE bit is “1” (parity error signal output enabled), the TXDi pin outputs “L” level if a parity error occurs while receiving data. To prevent this, set the UiERE bit after setting the UiMR register.

- Change the setting of bits SMD2 to SMD0 from “000_2” (serial I/O disabled) to “101_2” (UART mode transfer data length 8 bits).
- Change the setting of bits SMD2 to SMD0 from “001_2” (clock synchronous serial I/O mode) to “100_2” (UART mode transfer data length 7 bits).
- Change the setting of bits SMD2 to SMD0 from “001_2” (clock synchronous serial I/O mode) to “101_2” (UART mode transfer data length 8 bits).
- Change the setting of bits SMD2 to SMD0 from “001_2” (clock synchronous serial I/O mode) to “110_2” (UART mode transfer data length 9 bits).
- Change the setting of bits SMD2 to SMD0 from “010_2” (I^2C mode) to “101_2” (UART mode transfer data length 8 bits).

Note 1: If the pins are not used as CLKi or CTSi, these conditions apply when the pin level goes to “H”.

2.4.5 Operation of Serial I/O (transmission used for SIM interface)

In transmitting data in UARTi (i=0 to 3) mode (used for SIM interface), choose functions from those listed in Table 2.4.6. Operations of the circled items are described below. Figure 2.4.13 shows the operation timing, and Figures 2.4.14 and 2.4.15 show the set-up procedures.

Table 2.4.6. Choosed functions

Item	Set-up		Item	Set-up	
Transfer data format	O	Direct format	Transfer clock source	O	Internal clock (f_1/f_8/f_{32})
		Inverse format			External clock (CLKi pin)

Operation

(1) Setting the transmit enable bit and receive enable bit to “1” and writing transmission data to the UARTi (i=0 to 3) transmit buffer register readies the data transmissible status. Set UARTi (i=0 to 3) transfer interrupt for being enabled.

(2) Transmission data held in the UARTi (i=0 to 3) transmit buffer register is transmitted to the UARTi (i=0 to 3) transmit register. At this time, the first bit (the start bit) of the transmission data is transmitted from the TxDi (i=0 to 3) pin. Then, data is transmitted, bit by bit, in sequence: LSB, ⋯, MSB, parity bit, and stop bit(s).

(3) When the stop bit(s) is (are) transmitted, the transmit register empty flag goes to “1”, which indicates that transmission is completed. At this time, the UARTi (i=0 to 3) transmit interrupt request bit goes to “1”. The transfer clock stops at “H” level.

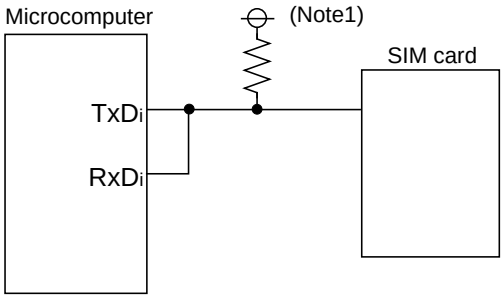
(4) If the transmission condition of the next data is ready when transmission is completed, a start bit is generated following to stop bit(s), and the next data is transmitted.

(5) If a parity error occurs, an L is output from the SIM card, and the RxDi (i=0 to 3) terminal turns to "L" level. Check the RxDi (i=0 to 3) terminal's level within the UARTi (i=0 to 3) transmission interrupt routine, and if it is found to be at the "L" level, then handle the error.

Note

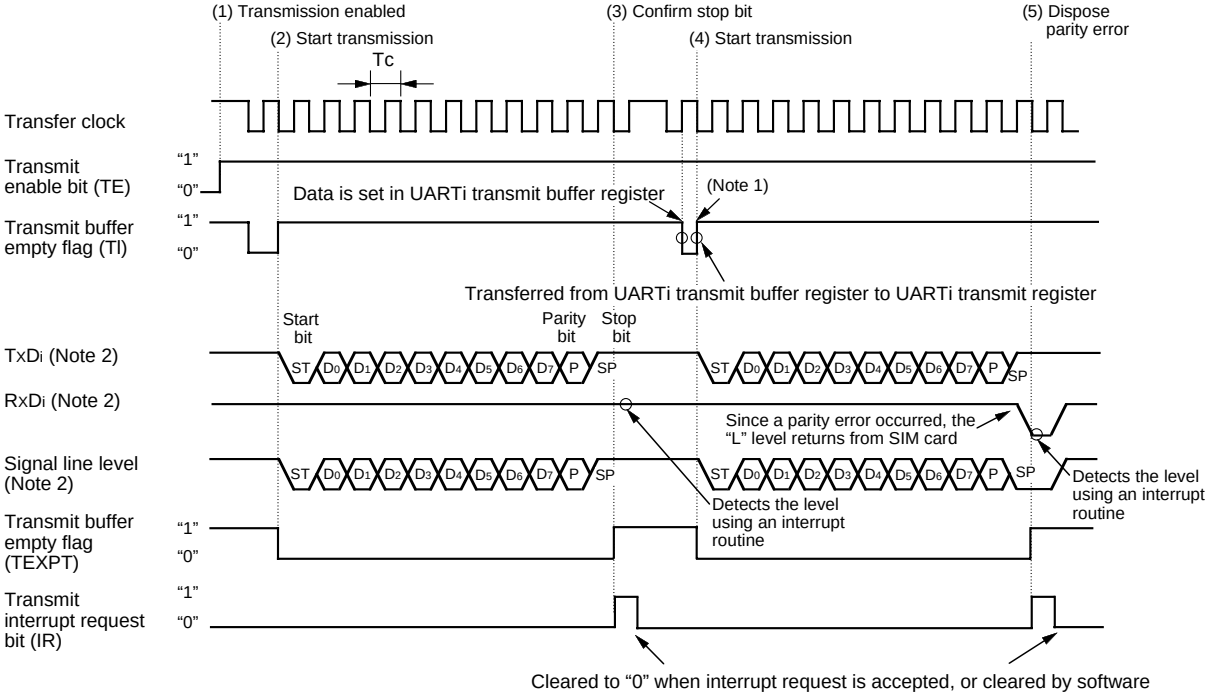
- The parity error level is determined within a UARTi (i=0 to 3) transmission interrupt. When a transmission interrupt request occurs, set the priority level of the transmission interrupt higher than those of other interrupts so that the interrupt routine can be immediately carried out. Either in the main routine or in an interrupt routine, the interrupt inhibition time has to be made as short as possible.
- Set the RxDi (i=0 to 3) pin's port direction register to input.
- Select N-channel open drain output for TxDi pin with data output select bit of UARTi (i=0 to 3) transmit/receive control register 0.

Example of wiring



Note1: TxDi pin is N-channel open drain and needs a pull-up resistance.
Note2: i=0 to 3

Example of operation (when direct format)


(1) Transmission enabled
(2) Start transmission
(3) Confirm stop bit
(4) Start transmission
(5) Dispose parity error
Tc
Transfer clock
Transmit enable bit (TE)
"1"
"0"
Data is set in UARTi transmit buffer register
(Note 1)
Transmit buffer empty flag (TI)
Transferred from UARTi transmit buffer register to UARTi transmit register
Start bit
Parity bit
Stop bit
TxDi (Note 2)
RxDi (Note 2)
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
Since a parity error occurred, the "L" level returns from SIM card
Signal line level (Note 2)
Detects the level using an interrupt routine
Transmit buffer empty flag (TEXPT)
Transmit interrupt request bit (IR)
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "1".

$Tc = 16 (n + 1) / fi$ or $16 (n + 1) / f_{EXT}$
fi : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

Note 1: The transmit is started with overflow timing of BRG after having written in a value at the transmit buffer in the above timing.
Note 2: TxDi and RxDi are connected in the manner of wired OR as shown in the connection diagram. Therefore, the signal levels of TxDi and RxDi should be the same, but the output signals are shown separately for ease of understanding. Also, the signal level resulting from connecting TxDi and RxDi is shown as a signal line level.
Note 3: i = 0 to 3

Figure 2.4.13. Operation timing of transmission in UART mode (used for SIM interface)

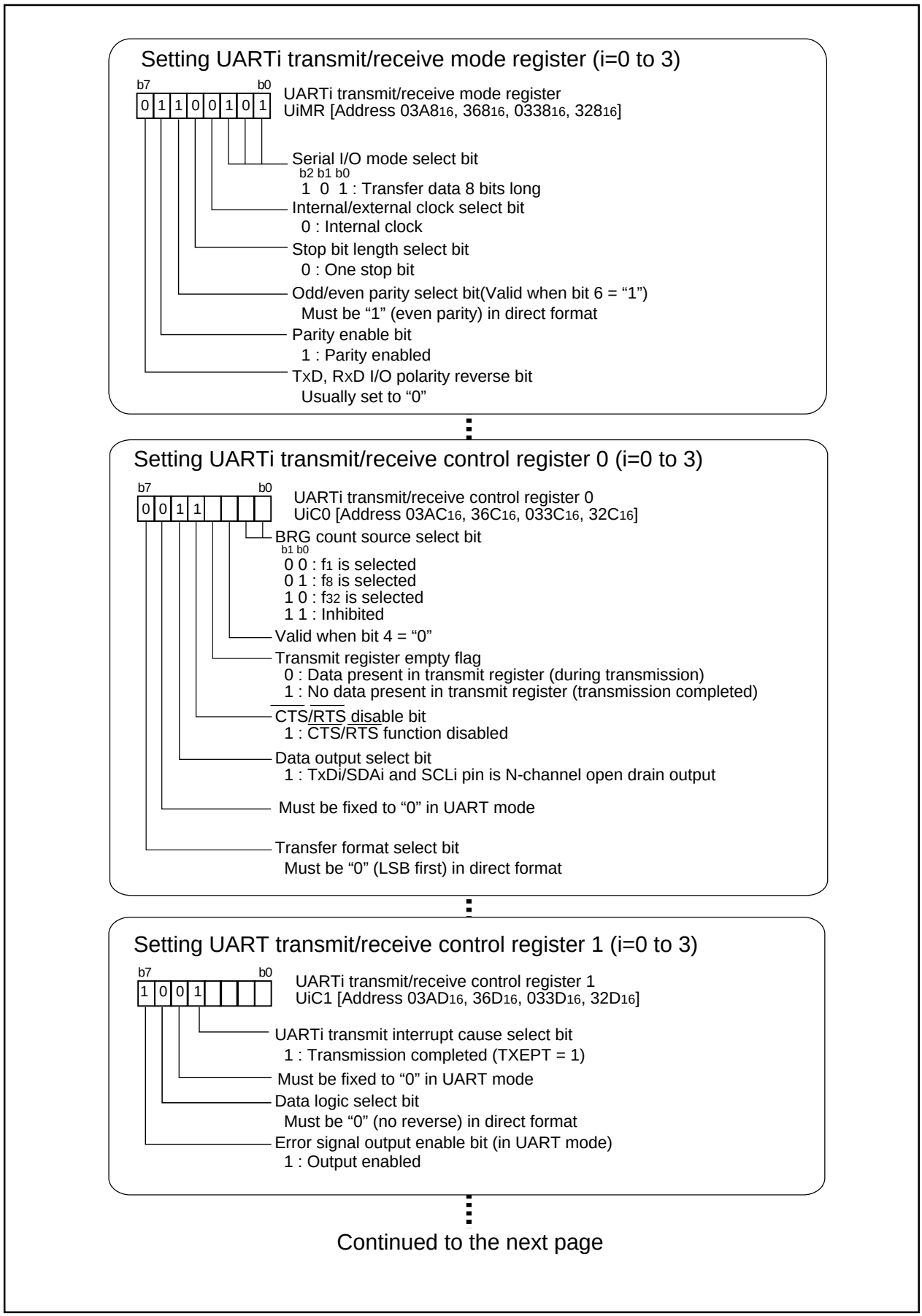

Setting UARTi transmit/receive mode register (i=0 to 3)
b7 b0
0 1 1 0 0 1 0 1
UARTi transmit/receive mode register
UiMR [Address 03A816, 36816, 033816, 32816]
Serial I/O mode select bit
b2 b1 b0
1 0 1 : Transfer data 8 bits long
Internal/external clock select bit
0 : Internal clock
Stop bit length select bit
0 : One stop bit
Odd/even parity select bit(Valid when bit 6 = "1")
Must be "1" (even parity) in direct format
Parity enable bit
1 : Parity enabled
TxD, RxD I/O polarity reverse bit
Usually set to "0"
Setting UARTi transmit/receive control register 0 (i=0 to 3)
b7 b0
0 0 1 1
UARTi transmit/receive control register 0
UiC0 [Address 03AC16, 36C16, 033C16, 32C16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : Inhibited
Valid when bit 4 = "0"
Transmit register empty flag
0 : Data present in transmit register (during transmission)
1 : No data present in transmit register (transmission completed)
CTS/RTS disable bit
1 : CTS/RTS function disabled
Data output select bit
1 : TxDi/SDAi and SCLi pin is N-channel open drain output
Must be fixed to "0" in UART mode
Transfer format select bit
Must be "0" (LSB first) in direct format
Setting UART transmit/receive control register 1 (i=0 to 3)
b7 b0
1 0 0 1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
UARTi transmit interrupt cause select bit
1 : Transmission completed (TXEPT = 1)
Must be fixed to "0" in UART mode
Data logic select bit
Must be "0" (no reverse) in direct format
Error signal output enable bit (in UART mode)
1 : Output enabled
Continued to the next page

Figure 2.4.14. Set-up procedure of transmission in UART mode (used for SIM interface) (1)

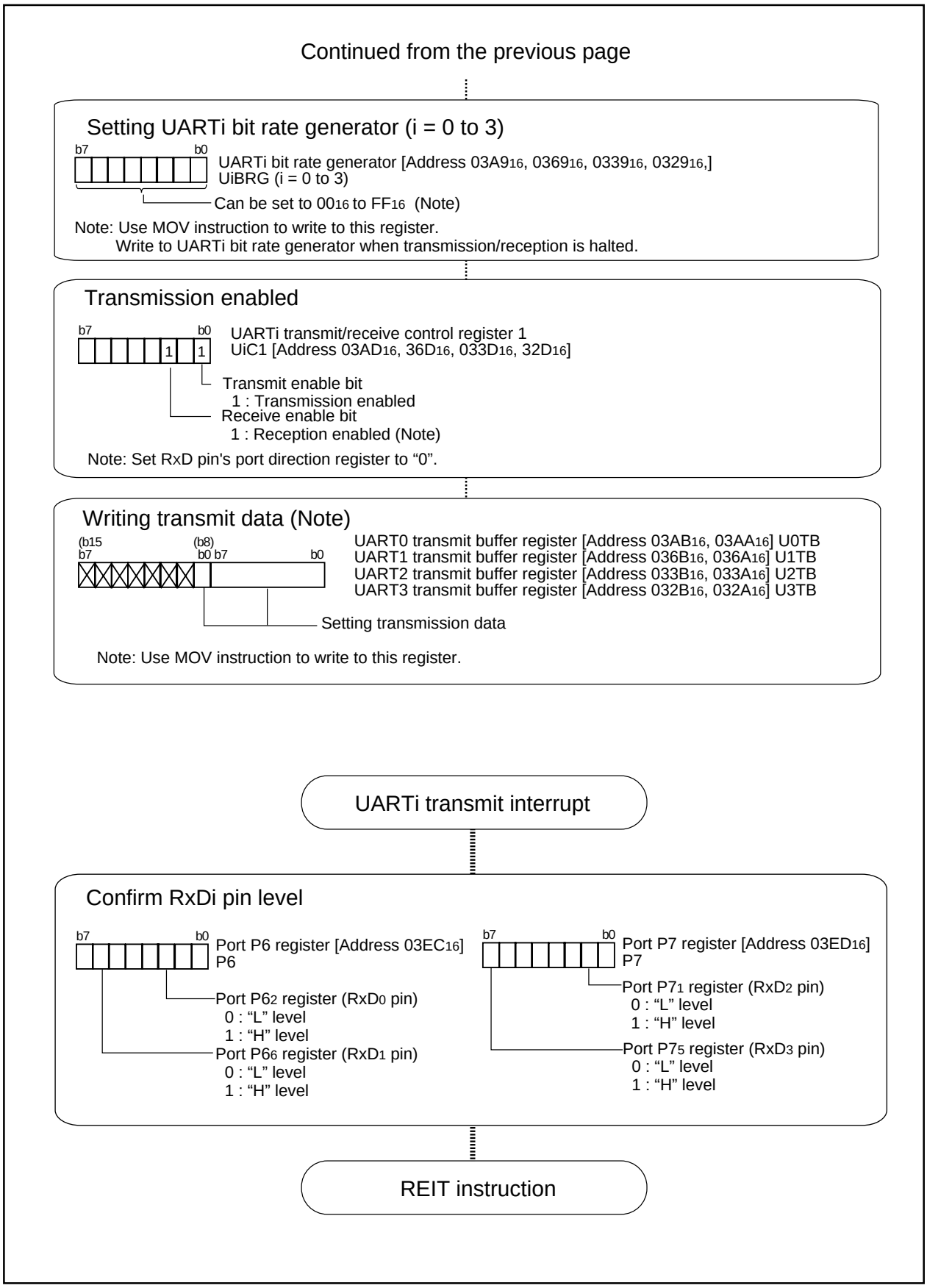

Continued from the previous page
Setting UARTi bit rate generator (i = 0 to 3)
b7
b0
UARTi bit rate generator [Address 03A916, 036916, 033916, 032916,]
UiBRG (i = 0 to 3)
Can be set to 0016 to FF16 (Note)
Note: Use MOV instruction to write to this register.
Write to UARTi bit rate generator when transmission/reception is halted.
Transmission enabled
b7
b0
1
1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit enable bit
1 : Transmission enabled
Receive enable bit
1 : Reception enabled (Note)
Note: Set RxD pin's port direction register to “0”.
Writing transmit data (Note)
(b15
(b8)
b7
b0 b7
b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting transmission data
Note: Use MOV instruction to write to this register.
UARTi transmit interrupt
Confirm RxDi pin level
b7
b0
Port P6 register [Address 03EC16]
P6
Port P62 register (RxD0 pin)
0 : “L” level
1 : “H” level
Port P66 register (RxD1 pin)
0 : “L” level
1 : “H” level
b7
b0
Port P7 register [Address 03ED16]
P7
Port P71 register (RxD2 pin)
0 : “L” level
1 : “H” level
Port P75 register (RxD3 pin)
0 : “L” level
1 : “H” level
REIT instruction

Figure 2.4.15. Set-up procedure of transmission in UART mode (used for SIM interface) (2)

2.4.6 Operation of Serial I/O (reception used for SIM interface)

In receiving data in UARTi (i=0 to 3) mode (used for SIM interface), choose functions from those listed in Table 2.4.7. Operations of the circled items are described below. Figure 2.4.16 shows the operation timing, and Figures 2.4.17 to 2.4.18 show the set-up procedures.

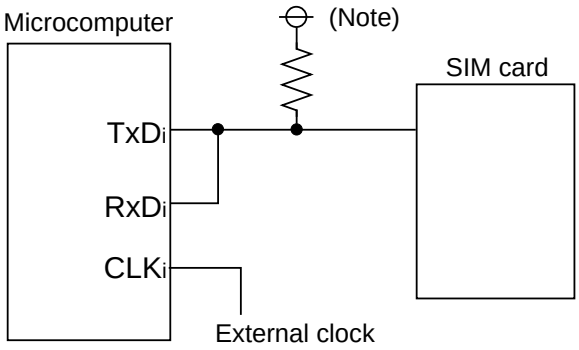
Table 2.4.7. Choosed functions

Item	Set-up		Item	Set-up	
Transfer data format		Direct format	Transfer clock source		Internal clock (f_1/f_8/f_{32})
	O	Inverse format		O	External clock (CLKi pin)

Operation
(1) Setting the transmit enable bit and receive enable bit to “1” readies data-receivable status.
(2) When the first bit (the start bit) of reception data is received from the RxDi (i=0 to 3) pin, data is received, bit by bit, in sequence: LSB, ⋯, MSB, and stop bit(s).
(3) When the stop bit(s) is (are) received, the content of the UARTi (i=0 to 3) receive register is transmitted to the UARTi (i=0 to 3) receive buffer register.
At this time, the receive complete flag goes to “1” to indicate that the reception is completed, and the UARTi (i=0 to 3) receive interrupt request bit goes to “1”.
(4) The receive complete flag goes to “0” when the lower-order byte of the UARTi (i=0 to 3) buffer register is read.
(5) When the parity error is occurred, TxDi (i=0 to 3) pin goes to “L” level.

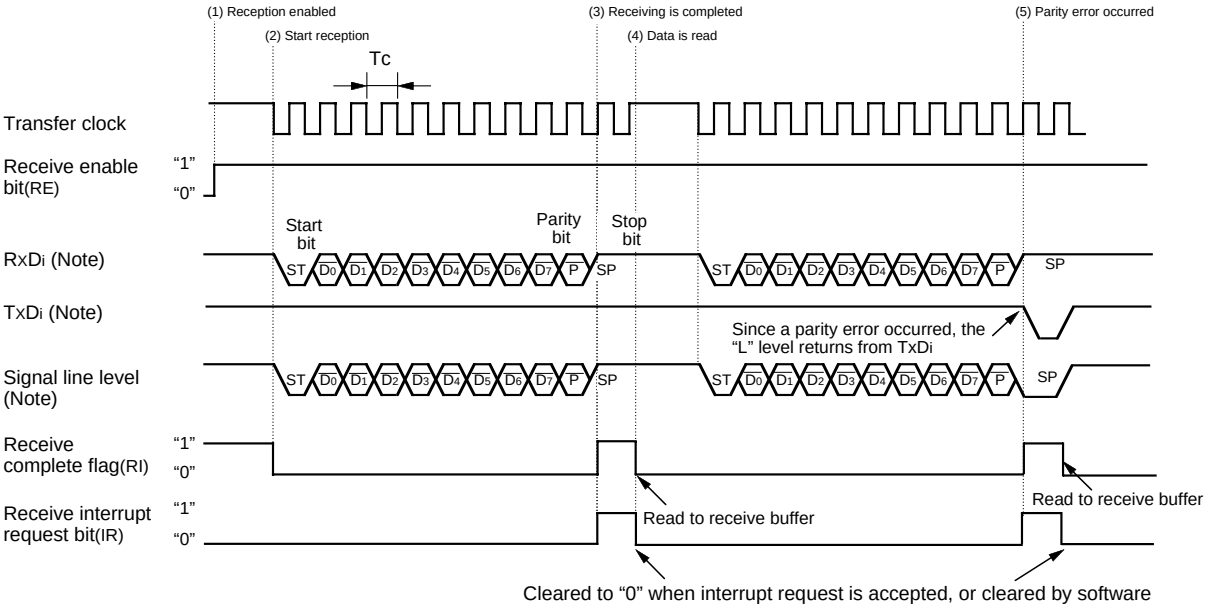
Note
• Set the RxDi and CLKi pins' port direction register to “0”.
• Select N-channel open drain output for TxDi (i=0 to 3) pin with data output select bit of UARTi (i=0 to 3) transmit/receive control register 0.

Example of wiring



Note1: TxDi pin is N-channel open drain and needs a pull-up resistance.
Note2: i=0 to 3

Example of operation (when inversed format)



Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fi or 16 (n + 1) / f_{EXT}
fi : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

Note : TxDi and RxDi are connected in the manner of wired OR as shown in the connection diagram. So TxDi and RxDi ought to become the same signal from the logical standpoint, but the output signals turn complex, so they are shown separately. Also, the signal level resulting from connecting TxDi and RxDi is shown as a signal line level.

Figure 2.4.16. Operation timing of reception in UART mode (used for SIM interface)

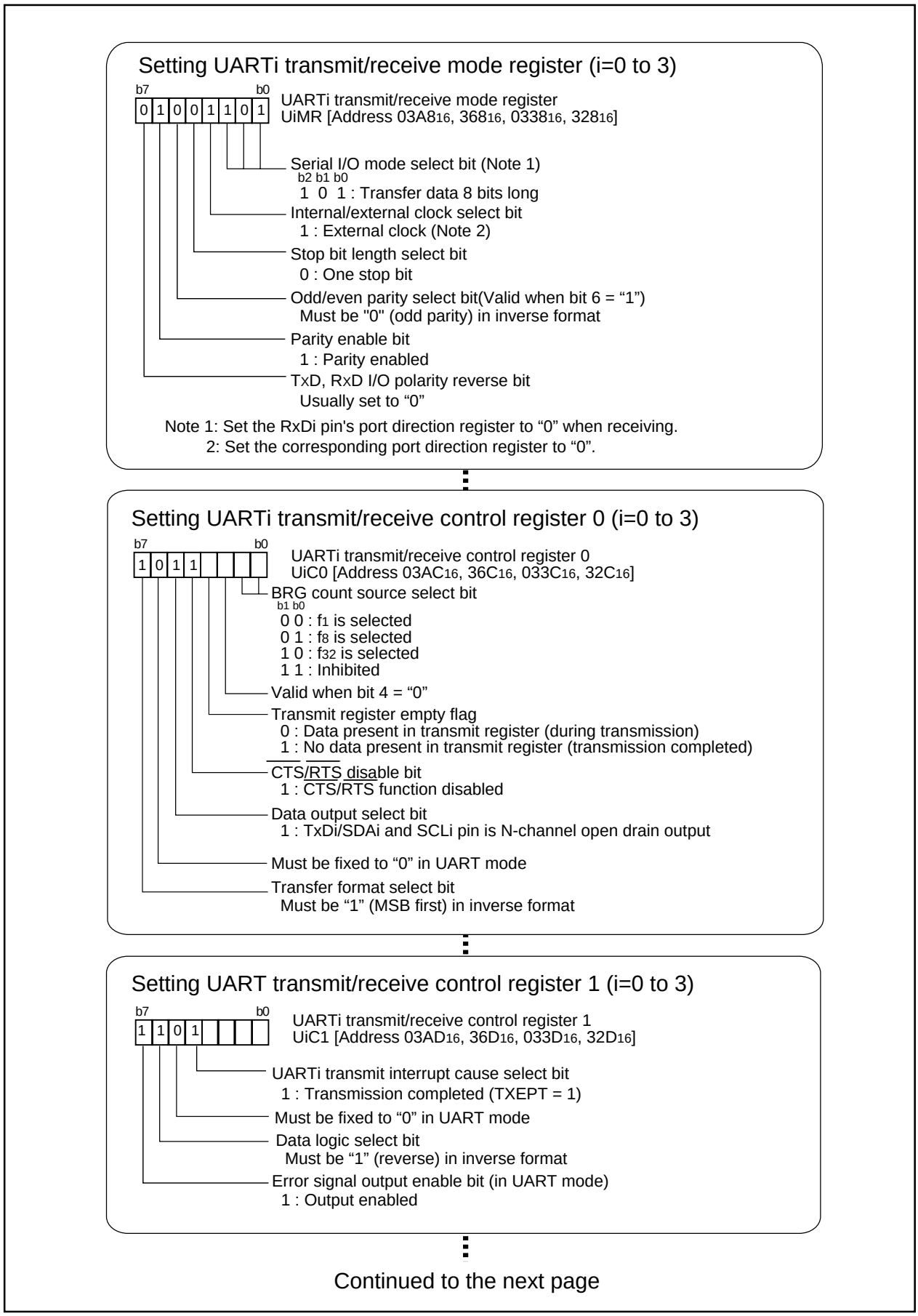

Setting UARTi transmit/receive mode register (i=0 to 3)
b7 b0
0 1 0 0 1 0 1
UARTi transmit/receive mode register
UiMR [Address 03A816, 36816, 033816, 32816]
Serial I/O mode select bit (Note 1)
b2 b1 b0
1 0 1 : Transfer data 8 bits long
Internal/external clock select bit
1 : External clock (Note 2)
Stop bit length select bit
0 : One stop bit
Odd/even parity select bit(Valid when bit 6 = "1")
Must be "0" (odd parity) in inverse format
Parity enable bit
1 : Parity enabled
TxD, RxD I/O polarity reverse bit
Usually set to "0"
Note 1: Set the RxDi pin's port direction register to "0" when receiving.
2: Set the corresponding port direction register to "0".
Setting UARTi transmit/receive control register 0 (i=0 to 3)
b7 b0
1 0 1 1
UARTi transmit/receive control register 0
UiC0 [Address 03AC16, 36C16, 033C16, 32C16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : Inhibited
Valid when bit 4 = "0"
Transmit register empty flag
0 : Data present in transmit register (during transmission)
1 : No data present in transmit register (transmission completed)
CTS/RTS disable bit
1 : CTS/RTS function disabled
Data output select bit
1 : TxDi/SDAi and SCLi pin is N-channel open drain output
Must be fixed to "0" in UART mode
Transfer format select bit
Must be "1" (MSB first) in inverse format
Setting UART transmit/receive control register 1 (i=0 to 3)
b7 b0
1 1 0 1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
UARTi transmit interrupt cause select bit
1 : Transmission completed (TXEPT = 1)
Must be fixed to "0" in UART mode
Data logic select bit
Must be "1" (reverse) in inverse format
Error signal output enable bit (in UART mode)
1 : Output enabled
Continued to the next page

Figure 2.4.17. Set-up procedure of reception in UART mode (used for SIM interface) (1)

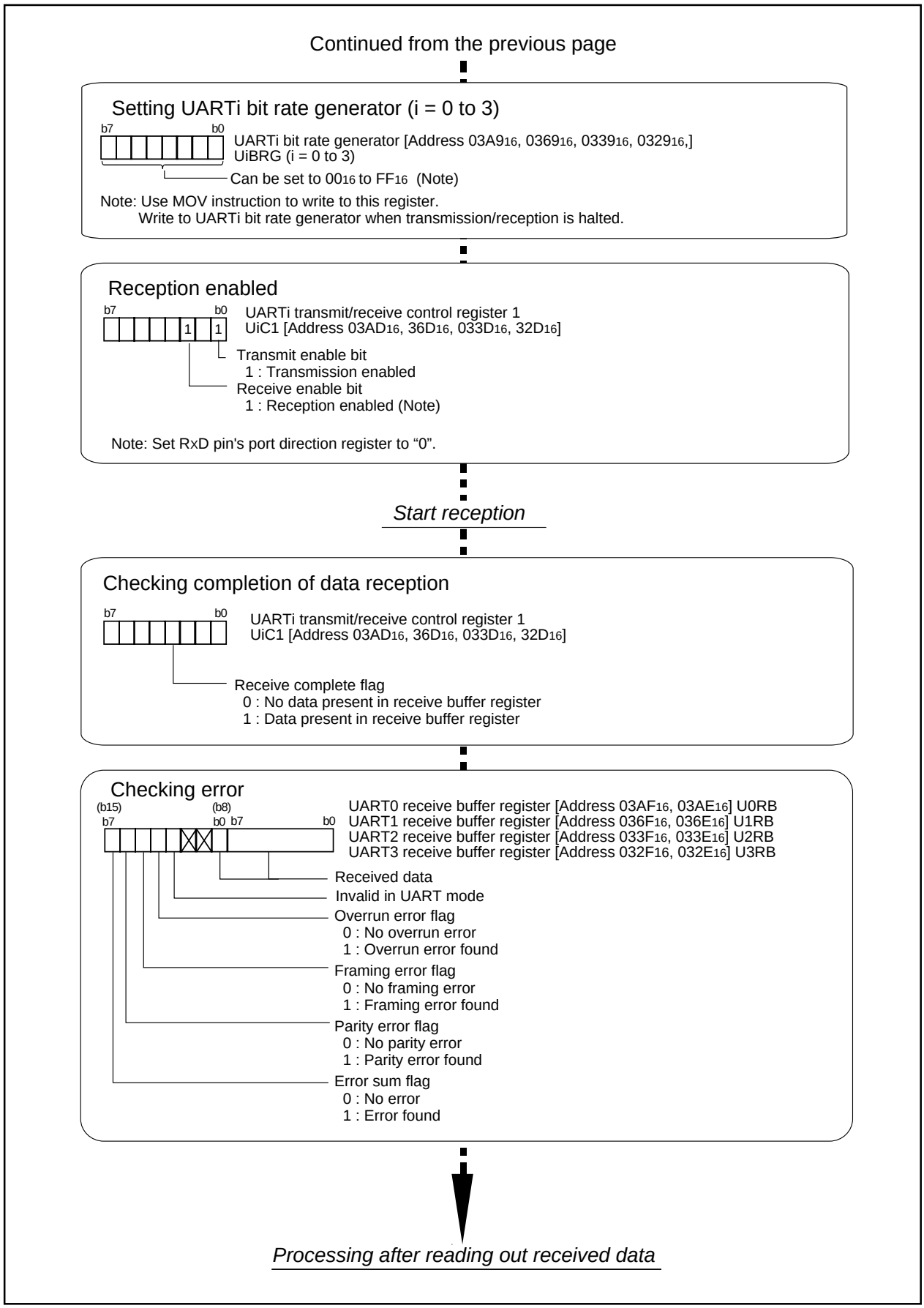

Continued from the previous page
Setting UARTi bit rate generator (i = 0 to 3)
b7 b0
UARTi bit rate generator [Address 03A916, 036916, 033916, 032916,]
UiBRG (i = 0 to 3)
Can be set to 0016 to FF16 (Note)
Note: Use MOV instruction to write to this register.
Write to UARTi bit rate generator when transmission/reception is halted.
Reception enabled
b7 b0
1 1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit enable bit
1 : Transmission enabled
Receive enable bit
1 : Reception enabled (Note)
Note: Set RxD pin's port direction register to “0”.
Start reception
Checking completion of data reception
b7 b0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Receive complete flag
0 : No data present in receive buffer register
1 : Data present in receive buffer register
Checking error
(b15) (b8)
b7 b0 b7 b0
UART0 receive buffer register [Address 03AF16, 03AE16] U0RB
UART1 receive buffer register [Address 036F16, 036E16] U1RB
UART2 receive buffer register [Address 033F16, 033E16] U2RB
UART3 receive buffer register [Address 032F16, 032E16] U3RB
Received data
Invalid in UART mode
Overrun error flag
0 : No overrun error
1 : Overrun error found
Framing error flag
0 : No framing error
1 : Framing error found
Parity error flag
0 : No parity error
1 : Parity error found
Error sum flag
0 : No error
1 : Error found
Processing after reading out received data

Figure 2.4.18. Set-up procedure of reception in UART mode (used for SIM interface)(2)

2.4.7 Clock Signals in used for the SIM Interface

In conforming to the SIM interface, the UART clock signal within the SIM card needs to conform to the UARTi (i=0 to 3) clock signal within the microprocessor. Two examples are given here as means of generating a UARTi clock signal within the microprocessor.

* In the case of setting a value equal to or less than (1/256 X 1/16) in the division rate of UARTi clock
  Choose f_1 for the UART's source clock signal and set an optional value in the bit rate generator.
* In the case of setting a value equal to or greater than (1/256 X 1/16) in the division rate of UARTi clock
  Set the bit rate generator to "0", turn the source clock signal to timer output and set an optional value in the timer.

Let F be the clock signal within the SIM card and D be the bit rate adjustment factor, then the formula for the UART clock signal becomes as follows. Figure 2.4.19 shows an example of connection.

- In the case of setting a value equal to or less than (1/256 X 1/16) in the division rate of UARTi clock
  UARTi clock signal within microprocessor = UART clock within SIM card

$$f_1 \times \frac{1}{\text{Bit rate generator} + 1} \times \frac{1}{16} = f_1 \times \frac{1}{\text{Timer Aj counter} + 1} \times \text{flip-flop} \times \frac{1}{F/D}$$

Let X_{IN} = 16 MHz, timer Aj counter = 1, F = 372, and D = 1, then the value to be set in the bit rate generator becomes

$$16 \times \frac{1}{\text{Bit rate generator} + 1} \times \frac{1}{16} = 16 \times \frac{1}{1+1} \times \frac{1}{2} \times \frac{1}{372/1}$$

Bit rate generator = 92

Table 2.4.8 shows an example of setting in the UARTi bit rate generator.

- In the case of setting a value equal to or greater than (1/256 X 1/16) in the division rate of UARTi clock
  UARTi clock signal within microprocessor = UART clock within SIM card

$$f_1 \times \frac{1}{\text{Timer Ak counter} + 1} \times \text{flip-flop} \times \frac{1}{\text{Bit rate generator} + 1} \times \frac{1}{16}$$

$$= f_1 \times \frac{1}{\text{Timer Aj counterr} + 1} \times \text{flip-flop} \times \frac{1}{F/D}$$

Let X_{IN}= 16 MHz, timer Aj counter = 3, bit rate generator = 0, F = 1860, and D = 1, then the value to be set in the timer Ak counter becomes

$$16 \times \frac{1}{\text{Timer Ak counter} + 1} \times \frac{1}{2} \times \frac{1}{0+1} \times \frac{1}{16} = 16 \times \frac{1}{3+1} \times \frac{1}{2} \times \frac{1}{1860/1}$$

Timer Ak counter = 464

Table 2.4.9 shows an example of setting in the timer Ak counter.

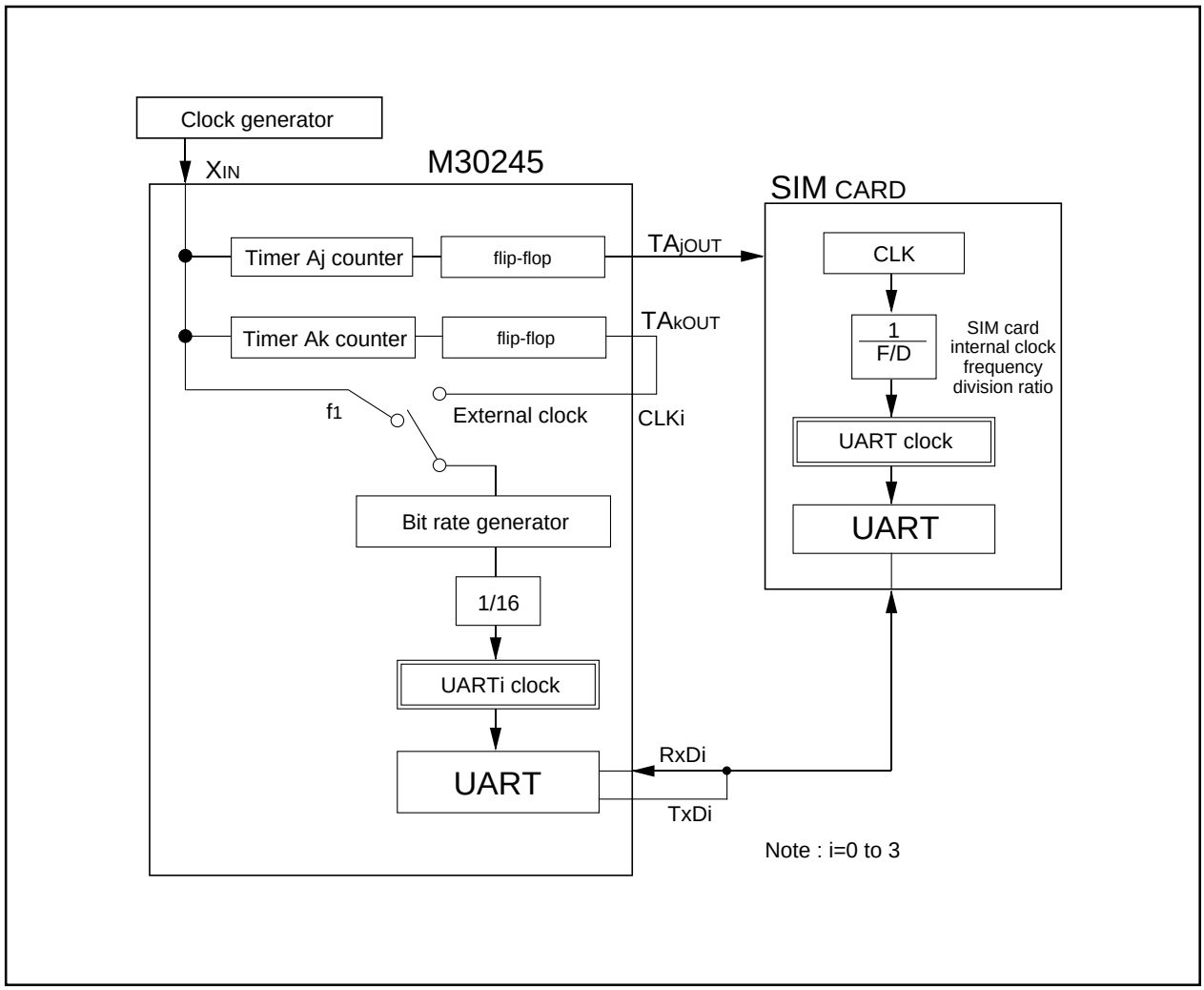


Figure 2.4.19. Example of connection

Table 2.4.8. UARTi bit rate adjustment factor (i=0 to 3)

SIM card internal clock F(Hz)	Bit rate D	F/D	UARTi bit rate generator set value	SIM card internal clock F(Hz)	Bit rate D	F/D	UARTi bit rate generator set value
372	1	372	92	1116	1	1116	
	2	186			2	558	
	4	93			4	279	
	8				8		
	16				16		
	1/2	744	185		1/2	2232	
	1/4	1488			1/4	4464	
	1/8	2976			1/8	8928	
	1/16	5952			1/16	17856	
	1/32	11904			1/32	35712	
	1/64	23808			1/64	71424	
558	1	558		1488	1	1488	
	2	279			2	744	185
	4				4	372	92
	8				8	186	
	16				16	93	
	1/2	1116			1/2	2976	
	1/4	2232			1/4	5952	
	1/8	4464			1/8	11904	
	1/16	8928			1/16	23808	
	1/32	17856			1/32	47616	
	1/64	35712			1/64	95232	
744	1	744	185	1860	1	1860	
	2	372	92		2	930	
	4	186			4	465	
	8	93			8		
	16				16		
	1/2	1488			1/2	3720	
	1/4	2976			1/4	7440	
	1/8	5952			1/8	14880	
	1/16	11904			1/16	29760	
	1/32	23808			1/32	59520	
	1/64	47616			1/64	119040	

Combination impossible to deal with due to the current specifications of M30245

Combination in which the F/D itself does not become an integer

Setting example under the following conditions.

f(XIN)=16 MHz

Timer Ak counter set value = 1

Table 2.4.9. TimerAi register adjustment factor

SIM card internal clock F(Hz)	Bit rate D	F/D	Timer Ai value	SIM card internal clock F(Hz)	Bit rate D	F/D	Timer Aj value
372	1	372	92	1116	1	1116	278
	2	186			2	558	
	4	93			4	279	
	8				8		
	16				16		
	1/2	744	185		1/2	2232	557
	1/4	1488	371		1/4	4464	1115
	1/8	2976	743		1/8	8928	2231
	1/16	5952	1487		1/16	17856	4463
	1/32	11904	2975		1/32	35712	8927
	1/64	23808	5951		1/64	71424	17855
558	1	558		1488	1	1488	371
	2	279			2	744	185
	4				4	372	92
	8				8	186	
	16				16	93	
	1/2	1116	278		1/2	2976	743
	1/4	2232	557		1/4	5952	1487
	1/8	4464	1115		1/8	11904	2975
	1/16	8928	2231		1/16	23808	5951
	1/32	17856	4463		1/32	47616	11903
	1/64	35712	8927		1/64	95232	23807
744	1	744	185	1860	1	1860	464
	2	372	92		2	930	
	4	186			4	465	
	8	93			8		
	16				16		
	1/2	1488	371		1/2	3720	929
	1/4	2976	743		1/4	7440	1859
	1/8	5952	1487		1/8	14880	3719
	1/16	11904	2975		1/16	29760	7439
	1/32	23808	5951		1/32	59520	14879
	1/64	47616	11903		1/64	119040	29759

Combination impossible to deal with due to the current specifications of M30245

Combination in which the F/D itself does not become an integer

Setting example under the following conditions.

f(XIN)=16 MHz

Timer Aj counter set value = 3, UARTi bit rate generator set value = 0

2.5 Serial Interface Special Function

2.5.1 Overview

Serial interface special function can control communications on the serial bus using $\overline{SSi}$ input pins. The following is an overview of the serial interface special function.

(1) Transmission/reception format

8-bit data

(2) Transfer rate

If the internal clock is selected as the transfer clock, the divide-by-2 frequency, resulting from the bit rate generator division, becomes the transfer rate. The bit rate generator count source can be selected from the following: f_1, f_8, and f_{32}. Clocks f_1, f_8, and f_{32} are derived by dividing the CPU's main clock by 1, 8, and 32 respectively.

Furthermore, if an external clock is selected as the transfer clock, the clock frequency input to the CLK pin becomes the transfer rate.

(3) Error detection

Fault error can be detected in the master mode.

When an "L" signal is input to an $\overline{SSi}$ pin in the multiple master system, it is judged there is another master existed, and the TxDi, RxDi and CLKi pins all become high impedance. Moreover, the fault error interrupt request bit becomes "1" and a fault error interrupt is generated.

(4) How to deal with an error

When the fault error flag is set to "0", output is restored to the clock output and data output pins. In the master mode, if an $\overline{SSi}$ input pin is H level, "0" can be written for the fault error flag. When an $\overline{SSi}$ input pin is L level, "0" cannot be written for the fault error flag. In the slave mode, the "0" can be written for the fault error flag regardless of the input to the $\overline{SSi}$ input pins.

(5) Function selection

For serial interface special function, the following functions can be selected:

(a) Function for choosing CLK polarity

This function switches the CLK polarity of the transfer clock. The following operations are available:

- Data is input at the falling edge of the transfer clock, and is output at the rising edge.
- Data is input at the rising edge of the transfer clock, and is output at the falling edge.

(b) Function for setting clock phase

This function switches the phase of the transfer clock. Choose either of the following:

- Without clock delay
- With clock delay

(c) Function for setting serial input pin

This function switches the serial bus control privilege between the master mode and slave mode. Choose either of the following:

- Master mode
- Slave mode

Following are some examples in which various functions (a) through (c) are selected:

- Transmission Operation WITH: outputting transmission data at falling edge of transfer clock, no clock delay, master mode
- Reception Operation WITH: inputting reception data at rising edge of transfer clock, clock delay, master mode
- Transmission Operation WITH: outputting transmission data at falling edge of transfer clock, no clock delay, slave mode
- Reception Operation WITH: inputting reception data at rising edge of transfer clock, clock delay, slave mode

(6) Input to the serial interface special function and the direction register

To input an external signal to the serial interface special function, set the direction register of the relevant port to input.

(7) Pins related to the serial interface special function

- CLK_0, CLK_1, CLK_2, CLK_3 pins Input/output pins for the transfer clock
- $RxD_0/SRxD_0$, $RxD_1/SRxD_1$, $RxD_2/SRxD_2$, $RxD_3/SRxD_3$ pins
  Input pins for data
- $TxD_0/STxD_0$, $TxD_1/STxD_1$, $TxD_2/STxD_2$, $TxD_3/STxD_3$ pins
  Output pins for data

(8) Registers related to the serial I/O

Figure 2.5.1 shows the memory map of serial interface special function-related registers, and Figures 2.5.2 to 2.5.7 show serial interface special function-related registers.

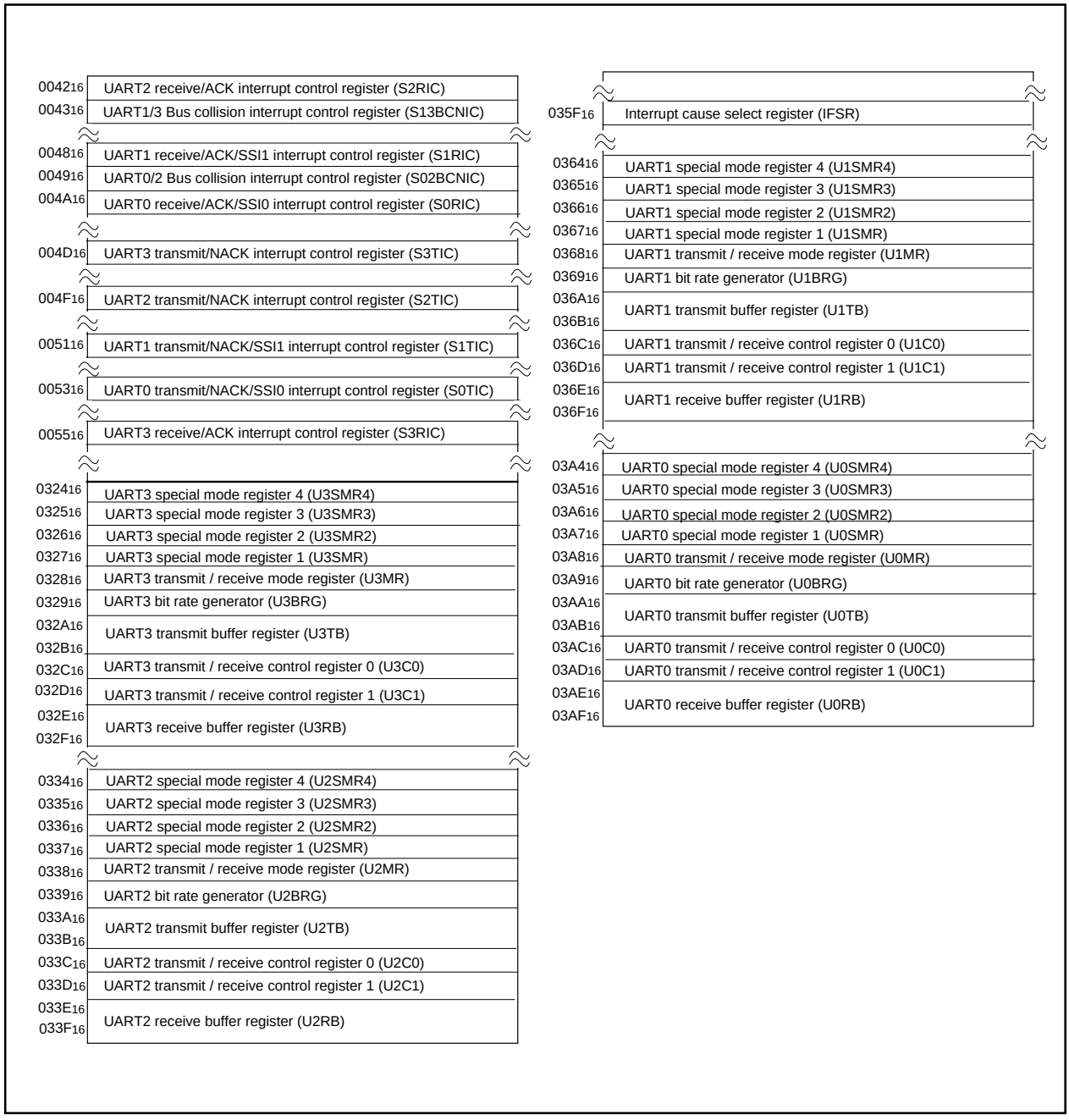
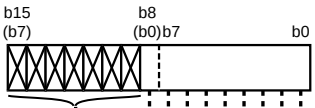

0042 UART2 receive/ACK interrupt control register (S2RIC)
0043 UART1/3 Bus collision interrupt control register (S13BCNIC)
0048 UART1 receive/ACK/SSI1 interrupt control register (S1RIC)
0049 UART0/2 Bus collision interrupt control register (S02BCNIC)
004A UART0 receive/ACK/SSI0 interrupt control register (S0RIC)
004D UART3 transmit/NACK interrupt control register (S3TIC)
004F UART2 transmit/NACK interrupt control register (S2TIC)
0051 UART1 transmit/NACK/SSI1 interrupt control register (S1TIC)
0053 UART0 transmit/NACK/SSI0 interrupt control register (S0TIC)
0055 UART3 receive/ACK interrupt control register (S3RIC)
0324 UART3 special mode register 4 (U3SMR4)
0325 UART3 special mode register 3 (U3SMR3)
0326 UART3 special mode register 2 (U3SMR2)
0327 UART3 special mode register 1 (U3SMR)
0328 UART3 transmit / receive mode register (U3MR)
0329 UART3 bit rate generator (U3BRG)
032A 032B UART3 transmit buffer register (U3TB)
032C UART3 transmit / receive control register 0 (U3C0)
032D UART3 transmit / receive control register 1 (U3C1)
032E 032F UART3 receive buffer register (U3RB)
0334 UART2 special mode register 4 (U2SMR4)
0335 UART2 special mode register 3 (U2SMR3)
0336 UART2 special mode register 2 (U2SMR2)
0337 UART2 special mode register 1 (U2SMR)
0338 UART2 transmit / receive mode register (U2MR)
0339 UART2 bit rate generator (U2BRG)
033A 033B UART2 transmit buffer register (U2TB)
033C UART2 transmit / receive control register 0 (U2C0)
033D UART2 transmit / receive control register 1 (U2C1)
033E 033F UART2 receive buffer register (U2RB)
035F Interrupt cause select register (IFSR)
0364 UART1 special mode register 4 (U1SMR4)
0365 UART1 special mode register 3 (U1SMR3)
0366 UART1 special mode register 2 (U1SMR2)
0367 UART1 special mode register 1 (U1SMR)
0368 UART1 transmit / receive mode register (U1MR)
0369 UART1 bit rate generator (U1BRG)
036A 036B UART1 transmit buffer register (U1TB)
036C UART1 transmit / receive control register 0 (U1C0)
036D UART1 transmit / receive control register 1 (U1C1)
036E 036F UART1 receive buffer register (U1RB)
03A4 UART0 special mode register 4 (U0SMR4)
03A5 UART0 special mode register 3 (U0SMR3)
03A6 UART0 special mode register 2 (U0SMR2)
03A7 UART0 special mode register 1 (U0SMR)
03A8 UART0 transmit / receive mode register (U0MR)
03A9 UART0 bit rate generator (U0BRG)
03AA 03AB UART0 transmit buffer register (U0TB)
03AC UART0 transmit / receive control register 0 (U0C0)
03AD UART0 transmit / receive control register 1 (U0C1)
03AE 03AF UART0 receive buffer register (U0RB)

Figure 2.5.1. Memory map of serial interface special function-related registers

UARTi transmit buffer register (i= 0 to 3) (Note)

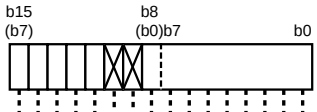


Symbol	Address	When reset
U0TB	$03AB_{16}$, $03AA_{16}$	Indeterminate
U1TB	$036B_{16}$, $036A_{16}$	Indeterminate
U2TB	$033B_{16}$, $033A_{16}$	Indeterminate
U3TB	$032B_{16}$, $032A_{16}$	Indeterminate

Bit Symbol	Function (clock synchronous serial I/O mode)	Function (UART mode)	R	W
———	Transmit data	Transmit data	×	○
———	———	Transmit data (9th bit)	×	○
———	Nothing is assigned. Write “0” when writing to these bits. The values are indeterminate when read.		–	–

Note: Use MOV instruction to write to this register.

UARTi receive buffer register (i= 0 to 3)



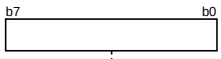
Symbol	Address	When reset
U0RB	$03AF_{16}$, $03AE_{16}$	Indeterminate
U1RB	$036F_{16}$, $036E_{16}$	Indeterminate
U2RB	$033F_{16}$, $033E_{16}$	Indeterminate
U3RB	$032F_{16}$, $032E_{16}$	Indeterminate

Bit Symbol	Bit Name	Function (clock synchronous serial I/O mode)	Function (UART mode)	R	W
———	———	Receive data	Receive data	○	×
———	———	———	Receive data (9th bit)	○	×
———	Nothing is assigned. Write “0” when writing to these bits. The values are indeterminate when read.			–	–
ABT	Arbitration lost detecting flag (Note 1)	0 : Not detected 1 : Detected	Invalid	○	○
OER	Overrun error flag (Note 2)	0 : No overrun error 1 : Overrun error	0 : No overrun error 1 : Overrun error	○	×
FER	Framing error flag (Note 2)	Invalid	0 : No framing error 1 : Framing error	○	×
PER	Parity error flag (Note 2)	Invalid	0 : No parity error 1 : Parity error	○	×
SUM	Error sum flag (Note 2)	Invalid	0 : No error 1 : Error	○	×

Note 1: Always write “0”.
Note 2: Bits 15 to 12 are set to “0000_2” when the serial I/O mode select bit (bits 0 to 2 at addresses $03A8_{16}$, 0368_{16}, 0338_{16}, 0328_{16}) are set to “000_2” or the receive enable bit is set to “0”. Bit 15 is set to “0” when all of bits 14 to 12 are set to “0”.
Bits 14 and 13 are also set to “0” when the lower byte of the UARTi receive buffer register (addresses $03AE_{16}$, $036E_{16}$, $033E_{16}$, $032E_{16}$) is read.

Figure 2.5.2. Serial interface special function-related registers (1)

UARTi bit rate generator (o=0 to 3) (Note 1, 2)

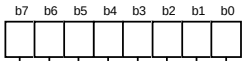


Symbol	Address	When reset
U0BRG	$03A9_{16}$	Indeterminate
U1BRG	0369_{16}	Indeterminate
U2BRG	0339_{16}	Indeterminate
U3BRG	0329_{16}	Indeterminate

Function	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}	–	O

Note 1: Use MOV instruction to write to this register.
Note 2: Write a value to this register while transmit/receive halts.

UARTi transmit/receive mode register (i = 0 to 3)



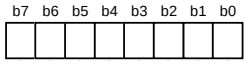
Symbol	Address	When reset
U0MR	$03A8_{16}$	00_{16}
U1MR	0368_{16}	00_{16}
U2MR	0338_{16}	00_{16}
U3MR	0328_{16}	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit (Note 3)	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Serial I/O mode 0 1 1 : I^2C mode Inhibited except in cases listed above	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long Inhibited except in cases listed above	O	O
SMD1				O	O
SMD2				O	O
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 1)	0 : Internal clock 1 : External clock (Note 1)	O	O
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	O	O
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	O	O
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	O	O
SLEP	TxD, RxD input/output polarity switch bit (Note 2)	0 : Normal 1 : Reversed		O	O

Note 1: When I^2C bus interface mode is selected, set the port direction register for the corresponding port (SCLi) to 0, or the port direction register to 1 and the port data register to 1. When a mode other than serial I/O mode is selected, set the port direction register for the corresponding port (CLKi) to 0.
Note 2: Normally set to “0”.
Note 3: Set the RxDi pin's port direction register to “0” when receiving.

Figure 2.5.3. Serial interface special function-related registers (2)

UARTi transmit/receive control register 0 (i=0 to 3)



Symbol: UiC0 (i=0 to 3) | Address: 03AC16, 036C16, 033C16, 032C16, | When reset: 0816

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f1 is selected 0 1 : f8 is selected 1 0 : f32 is selected 1 1 : Inhibited	b1 b0 **0 0 : f1 is selected** **0 1 : f8 is selected** **1 0 : f32 is selected** **1 1 : Inhibited**	○	○
CLK1				○	○
CRS	$\overline{CTS}/\overline{RTS}$ function select bit	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 4)	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 4)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	$\overline{CTS}/\overline{RTS}$ disable bit	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : $\overline{CTS}/\overline{RTS}$ function disabled	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : $\overline{CTS}/\overline{RTS}$ function disabled	○	○
NCH (Note 2)	Data output select bit	0 : TxDi/SDAi and SCLi pin is CMOS output 1 : TxDi/SDAi and SCLi pin is N-channel open drain output	0 : TxDi/SDAi and SCLi pin is CMOS output 1 : TxDi/SDAi and SCLi pin is N-channel open drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Set to "0"	○	○
UFORM	Transfer format select bit (Note 3)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

Note 1: Set the corresponding port direction register to "0".
Note 2: UART2 transfer pin (TxD2: P70 and SCL2: P71) is N-channel open drain output. It cannot be set to CMOS output.
Note 3: Only clock synchronous serial I/O mode and 8-bit UART mode are valid.
Note 4: The corresponding port register and port direction register are invalid.

UARTi transmit/receive control register 1 (i= 0 to 3)

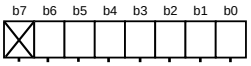
Symbol: UiC1 (i=0 to 3) | Address: 03AD16, 036D16, 033D16, 032D16 | When reset: 0216

Bit Symbol	Bit Name	Function (clock synchronous serial I/O mode)	Function (UART mode)	R	W
TE	Transmit enable bit	0 : Transmit disabled 1 : Transmit enabled		○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register		○	○
RE	Receive enable bit	0 : Receive disabled 1 : Receive enabled		○	○
RI	Receive complete flag	0 : Data packet in receive buffer register 1 : No data packet in receive buffer register		○	○
UiIRS	UARTi transmit interrupt cause select bit	0 : Transmit buffer empty (TI =1) 1 : Transmit buffer completed (TXEPT =1)		○	○
UiRRM	UARTi continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Set to "0"	○	○
UiLCH	Data logic select bit	0 : No reverse 1 : Reverse		○	○
UiERE	Error signal output enable bit	Set to "0" The value is indeterminate when read.	0 : Output disabled 1 : Output enabled (Note 1)	○	○

Note 1: When disabling the error signal output, set the UiERE bit to "0" after setting the UiMR register.

Figure 2.5.4. Serial interface special function-related registers (3)

UARTi special mode register 1 (i= 0 to 3)

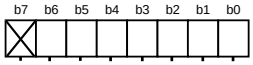


Symbol	Address	When reset
UiSMR (i=0 to 3)	$03A7_{16}$, 0367_{16}, 0337_{16}, 0327_{16}	00_{16}

Bit Symbol	Bit Name	Function (clock synchronous serial I/O mode)	Function (UART mode)	R	W
IICM	I^2C mode select bit	0 : Normal mode 1 : I^2C mode	Set to “0”	○	○
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	Set to “0”	○	○
BBS	Bus busy flag	0 : STOP detected 1 : START detected	Set to “0”	○	○
LSYN	SCLL sync output enable bit	0 : Disabled 1 : Enabled	Set to “0”	○	○ (Note 1)
ABSCS	Bus collision detect sampling clock select bit	Set to “0”	0 : Rising edge of transfer clock 1 : Timer Ai underflow signal (Note 2)	○	○
ACSE	Auto-clear function select bit of transmit enable bit	Set to “0”	0 : No auto clear function 1 : Auto clear when bus collision occurs	○	○
SSS	Transmit start condition select bit	Set to “0”	0 : Ordinary 1 : Falling edge of RxDi	○	○
——	Nothing is assigned. Write "0" when writing to this bit. The values are indeterminate when read.			—	—

Note 1: Only “0“ may be written
Note 2: UART0 Timer A3 underflow signal, UART1: Timer A4 underlfow signal, UART2 :Timer A0 underflow signal.

UARTi special mode register 2 (i= 0 to 3)



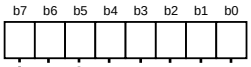
Symbol	Address	When reset
UiSMR2 (i=0 to 3)	$03A6_{16}$, 0366_{16}, 0336_{16}, 0326_{16}	00_{16}

Bit Symbol	Bit Name	Function	R	W
IICM2	I^2C mode select bit 2	0 : NACK/ACK interrupt (DMA source-ACK) Transfer to receive buffer at the rising edge of last bit of receive clock. Receive interrupt occurs at the rising edge of last bit of receive clock. 1 : UART transfer/receive interrupt (DMA source-UART receive) Transfer to receive buffer at the falling edge of last bit of receive clock. Receive interrupt occurs at the falling edge of last bit of receive clock	○	○
CSC	Clock synchronous bit	0 : Disable 1 : Enable	○	○
SWC	SCL wait output bit	0 : Disable 1 : Enable (Note 1)	○	○
ALS	SDA output stop bit	0 : Disable 1 : Enable	○	○
STC	UARTi initialize bit	0 : Disable 1 : Enable (Note 1)	○	○
SWC2	SCL Wait output bit 2	0 : UARTi clock 1 : 0 output (Note 1)	○	○
SDHI	SDA output inhibit bit	0 : Disabled 1 : Enabled (high impedance)	○	○
——	Nothing is assigned. Write "0" when writing to this bit. The values are indeterminate when read.		—	—

Note 1: These bits are unavailable when SCLi is external clock.

Figure 2.5.5. Serial interface special function-related registers (4)

UARTi special mode register 3 (i= 0 to 3)



Symbol	Address	When reset
UiSMR3 (i=0 to 3)	03A5$_{16}$, 0365$_{16}$, 0335$_{16}$, 0325$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	R	W
SSE	$\overline{SS}$ port function enable bit (Note 1)	0 : $\overline{SS}$ function disabled 1 : $\overline{SS}$ function enabled	○	○
CKPH	Clock phase set bit	0 : No clock delay 1 : Clock delay	○	○
DINC	Serial input port set bit	0 : Select TxDi and RxDi (master mode) 1 : Select STxDi and SRxDi (slave mode)	○	○
NODC	Clock output select bit	0 : CLKi is CMOS output 1 : CLKi is N-channel open drain output	○	○
ERR	Fault error flag	0 : No fault error 1 : Fault error (Note 2)	○	○
DL0	SDA (TxDi) digital delay time set bit (Notes 3,4)	b7 b6 b5 0 0 0 : No delay 0 0 1 : 1 to 2-cycle of UiBRG count source 0 1 0 : 2 to 3-cycle of UiBRG count source 0 1 1 : 3 to 4-cycle of UiBRG count source 1 0 0 : 4 to 5-cycle of UiBRG count source 1 0 1 : 5 to 6-cycle of UiBRG count source 1 1 0 : 6 to 7-cycle of UiBRG count source 1 1 1 : 7 to 8-cycle of UiBRG count source	○	○
DL1			○	○
DL2			○	○

Note 1: Set $\overline{SS}$ function after setting $\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 of UARTi transfer/receive control register 0) to "1"
Note 2: Only "0" may be written.
Note 3: These bits are used for SDAi (TxDi) output digital delay when using UARTi for I^2C interface. Otherwis set to "000".
Note 4: The amount of delay varies with the load on SCLi and SDAi pins.
When external clock is selected, delay is increased by approximately 100ns.

Figure 2.5.6. Serial interface special function-related registers (5)

UARTi special mode register 4 (i= 0 to 3)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
UiSMR4 (i=0 to 3)	$03A4_{16}$, 0364_{16}, 0334_{16}, 0324_{16}	00_{16}

Bit Symbol	Bit Name	Function	R	W
STAREQ	Start condition generate bit (Note 1)	0 : Clear 1 : Start	○	○
RSTAREQ	Restart condition generate bit (Note 1)	0 : Clear 1 : Start	○	○
STPREQ	Stop condition generate bit (Note 1)	0 : Clear 1 : Start	○	○
STSPSEL	SCL, SDA output select bit	0 : Ordinal block 1 : Start/stop condition generate block	○	○
ACKD	ACK data bit	0 : ACK 1 : NACK	○	○
ACKC	ACK data output enable bit	0 : SI/O data output 1 : ACKD output	○	○
SCLHI	SCL output stop enable bit	0 : Disabled 1 : Enabled	○	○
SWC9	SCL wait output bit 3	0 : SCL “L” hold disabled 1 : SCL “L” hold enabled (Note 2)	○	○

Note 1: These bits automatically become “0” when a start condition is generated.
Note 2: This bit is unavailable when SCLi is external clock.

Interrupt request cause select register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
IFSR	$035F_{16}$,	00_{16}

Bit Symbol	Bit name	Function	R	W
IFSR0	INT0 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
IFSR1	INT1 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
IFSR2	INT2 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
Nothing is assigned. Write “0” when writing to this bit. The value is indeterminate when read.			—	—
IFSR6	Bus collision interrupt request cause select bit 0	0 : UART0 Bus collision / Start/stop condition detection / Trouble error detection 1 : UART2 Bus collision / Start/stop condition detection / Trouble error detection	○	○
IFSR7	Bus collision interrupt request cause select bit 1	0 : UART1 Bus collision / Start/stop condition detection / Trouble error detection 1 : UART3 Bus collision / Start/stop condition detection / Trouble error detection	○	○

Figure 2.5.7. Serial interface special function-related registers (6)

2.5.2 Operation of Serial Interface Special Function (transmission in master mode without delay)

In transmitting data in serial interface special function master mode, choose functions from those listed in Table 2.5.1. Operations of the circled items are described below. Figure 2.5.8 shows the operation timing, and Figures 2.5.9 and 2.5.10 show the set-up procedures.

Table 2.5.1. Choosed functions

Item	Set-up		Item	Set-up	
Transfer clock source	O	Internal clock (f_1 / f_8 / f_{32})	$\overline{SSi}$ port function enable		$\overline{SSi}$ function disabled
		External clock (CLKi pin)		O	$\overline{SSi}$ function enabled
CLK polarity	O	Output transmission data at the falling edge of the transfer clock	Clock phase set	O	Without clock delay
					With clock delay
		Output transmission data at the rising edge of the transfer clock	Serial input port set	O	TxDi, RxDi selected (master mode)
Transmission interrupt factor	O	Transmission buffer empty			STxDi, SRxDi selected (slave mode)
		Transmission complete			

Operation (1) Set an $\overline{SS}$ port of the receiver side IC to output "L" level.

(2) Setting the transmit enable bit to “1” and writing transmission data to the UARTi transmit buffer register makes data transmissible status ready.

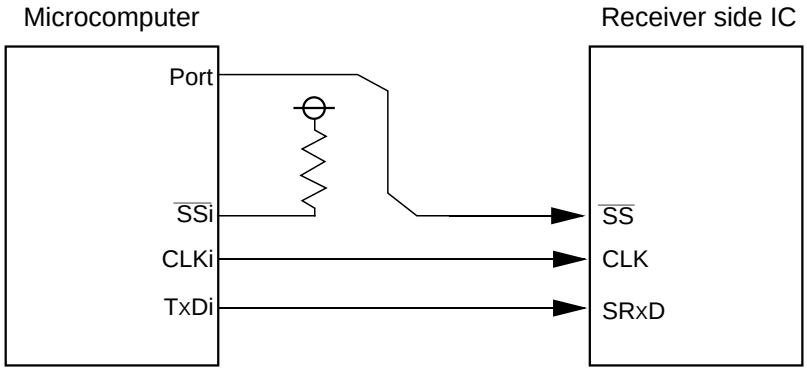
(3) In synchronization with the first falling edge of the transfer clock, transmission data held in the UARTi transmit buffer register is transmitted to the UARTi transmit register. At this time, the UARTi transmit interrupt request bit goes to “1”. Also, the first bit of the transmission data is transmitted from the TxDi pin. Then the data is transmitted bit by bit from the lower order in synchronization with the falling edges.

(4) When transmission of 1-byte data is completed, the transmit register empty flag goes to “1”, which indicates that transmission is completed. The transfer clock stops at “L” level.

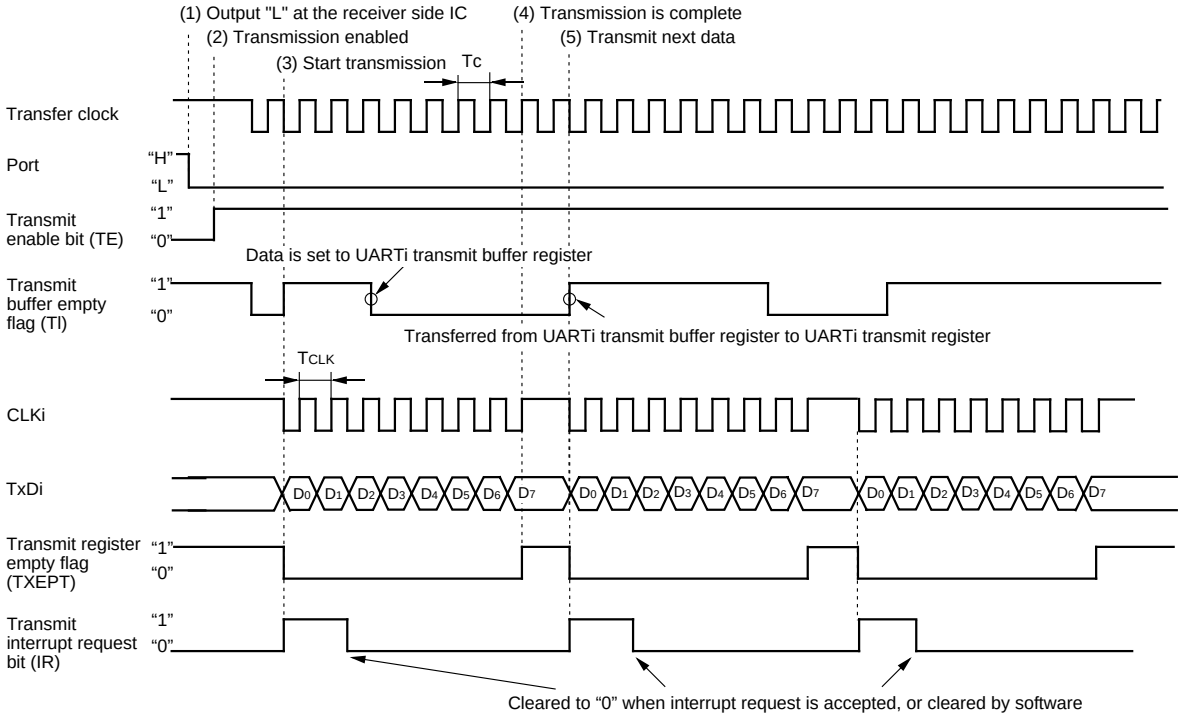
(5) If the next transmission data is set in the UARTi transmit buffer register while transmission is in progress (before the eighth bit has been transmitted), the data is transmitted in succession.

Note • Set $\overline{SSi}$ pin to "H" level. If "L" level is input to the pin, a fault error will be generated.

Example of wiring



Example of operation



Shown in ( ) are bit symbols.
The above timing applies to the following settings:
- Internal clock is selected.
- CLK polarity select bit = "0".
- Transmit interrupt cause select bit = "0".

$Tc = T_{CLK} = 2(n + 1) / fi$
fi: frequency of BRGi count source (f_1, f_8, f_{32})
n: value set to BRGi

Figure 2.5.8. Operation timing of transmission in serial interface special function master mode, without clock delay

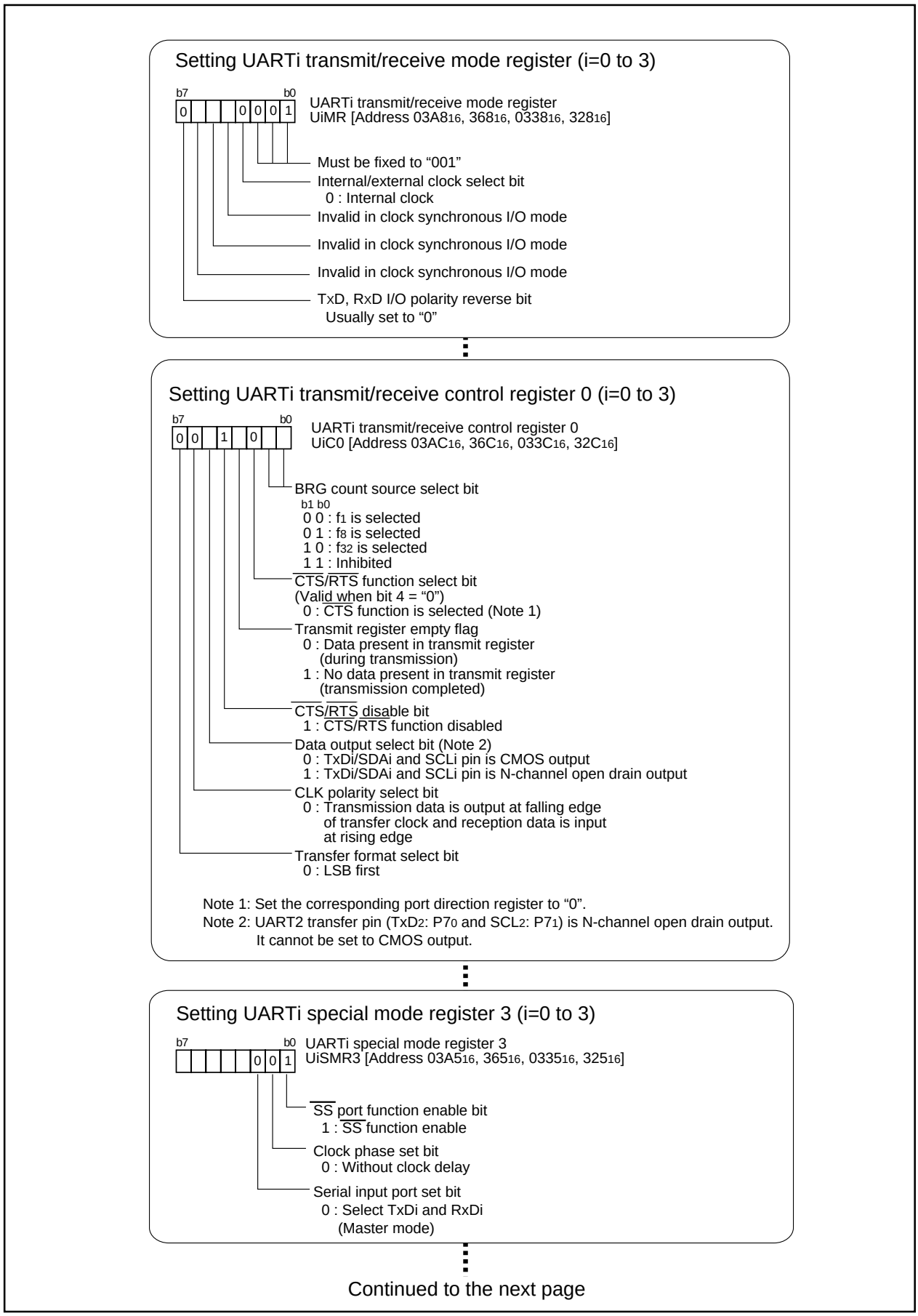

Setting UARTi transmit/receive mode register (i=0 to 3)
b7 b0
0 0 0 0 1
UARTi transmit/receive mode register
UiMR [Address 03A816, 36816, 033816, 32816]
Must be fixed to "001"
Internal/external clock select bit
0 : Internal clock
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
TxD, RxD I/O polarity reverse bit
Usually set to "0"
Setting UARTi transmit/receive control register 0 (i=0 to 3)
b7 b0
0 0 1 0
UARTi transmit/receive control register 0
UiC0 [Address 03AC16, 36C16, 033C16, 32C16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : Inhibited
CTS/RTS function select bit
(Valid when bit 4 = "0")
0 : CTS function is selected (Note 1)
Transmit register empty flag
0 : Data present in transmit register
(during transmission)
1 : No data present in transmit register
(transmission completed)
CTS/RTS disable bit
1 : CTS/RTS function disabled
Data output select bit (Note 2)
0 : TxDi/SDAi and SCLi pin is CMOS output
1 : TxDi/SDAi and SCLi pin is N-channel open drain output
CLK polarity select bit
0 : Transmission data is output at falling edge
of transfer clock and reception data is input
at rising edge
Transfer format select bit
0 : LSB first
Note 1: Set the corresponding port direction register to "0".
Note 2: UART2 transfer pin (TxD2: P70 and SCL2: P71) is N-channel open drain output.
It cannot be set to CMOS output.
Setting UARTi special mode register 3 (i=0 to 3)
b7 b0
0 0 1
UARTi special mode register 3
UiSMR3 [Address 03A516, 36516, 033516, 32516]
SS port function enable bit
1 : SS function enable
Clock phase set bit
0 : Without clock delay
Serial input port set bit
0 : Select TxDi and RxDi
(Master mode)
Continued to the next page

Figure 2.5.9. Set-up procedure of transmission in serial interface special function master mode, without clock delay (1)

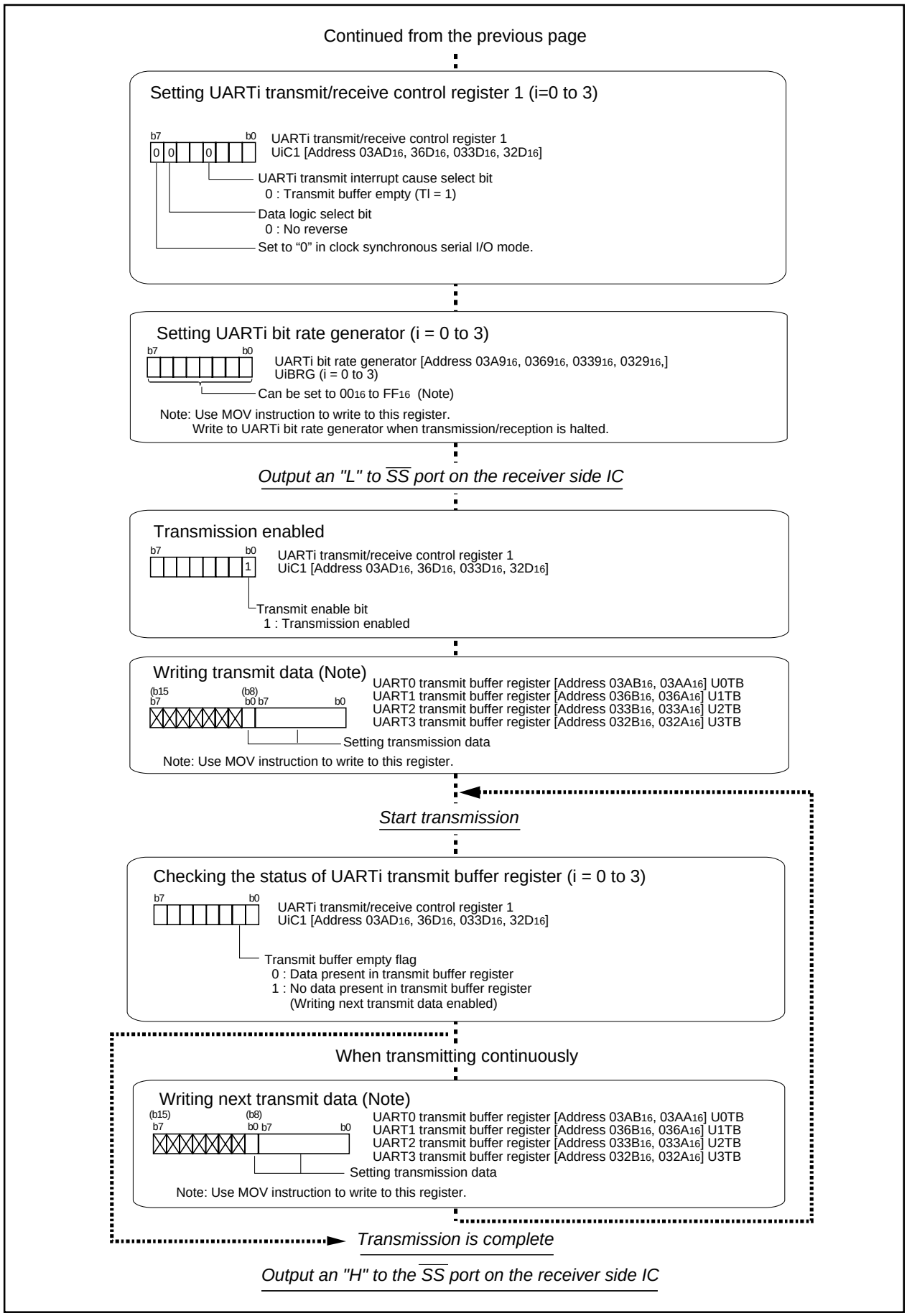

Continued from the previous page
Setting UARTi transmit/receive control register 1 (i=0 to 3)
b7 b0
0 0 0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
UARTi transmit interrupt cause select bit
0 : Transmit buffer empty (TI = 1)
Data logic select bit
0 : No reverse
Set to "0" in clock synchronous serial I/O mode.
Setting UARTi bit rate generator (i = 0 to 3)
b7 b0
UARTi bit rate generator [Address 03A916, 036916, 033916, 032916,]
UiBRG (i = 0 to 3)
Can be set to 0016 to FF16 (Note)
Note: Use MOV instruction to write to this register.
Write to UARTi bit rate generator when transmission/reception is halted.
Output an "L" to SS port on the receiver side IC
Transmission enabled
b7 b0
1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit enable bit
1 : Transmission enabled
Writing transmit data (Note)
(b15) (b8)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting transmission data
Note: Use MOV instruction to write to this register.
Start transmission
Checking the status of UARTi transmit buffer register (i = 0 to 3)
b7 b0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit buffer empty flag
0 : Data present in transmit buffer register
1 : No data present in transmit buffer register
(Writing next transmit data enabled)
When transmitting continuously
Writing next transmit data (Note)
(b15) (b8)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting transmission data
Note: Use MOV instruction to write to this register.
Transmission is complete
Output an "H" to the SS port on the receiver side IC

Figure 2.5.10. Set-up procedure of transmission in serial interface special function master mode, without clock delay (2)

2.5.3 Operation of Serial Interface Special Function (reception in master mode with clock delay)

In receiving data in serial interface special function master mode, choose functions from those listed in Table 2.5.2. Operations of the circled items are described below. Figure 2.5.11 shows the operation timing, and Figures 2.5.12 and 2.5.13 show the set-up procedures.

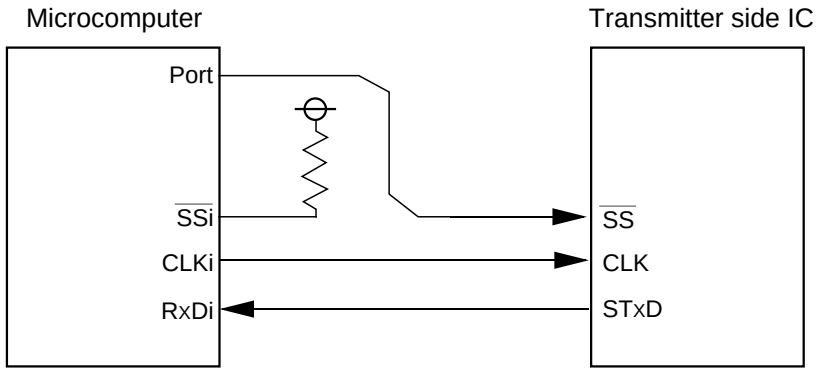
Table 2.5.2. Choosed functions

Item	Set-up		Item	Set-up	
Transfer clock source	O	Internal clock (f_1 / f_8 / f_{32})	$\overline{SSi}$ port function enable		$\overline{SSi}$ function disabled
		External clock (CLKi pin)		O	$\overline{SSi}$ function enabled
CLK polarity	O	Output reception data at the rising edge of the transfer clock	Clock phase set		Without clock delay
				O	With clock delay
		Output reception data at the falling edge of the transfer clock	Serial input port set	O	TxDi, RxDi selected (master mode)
Continuous receive mode	O	Disabled			STxDi, SRxDi selected (slave mode)
		Enabled			

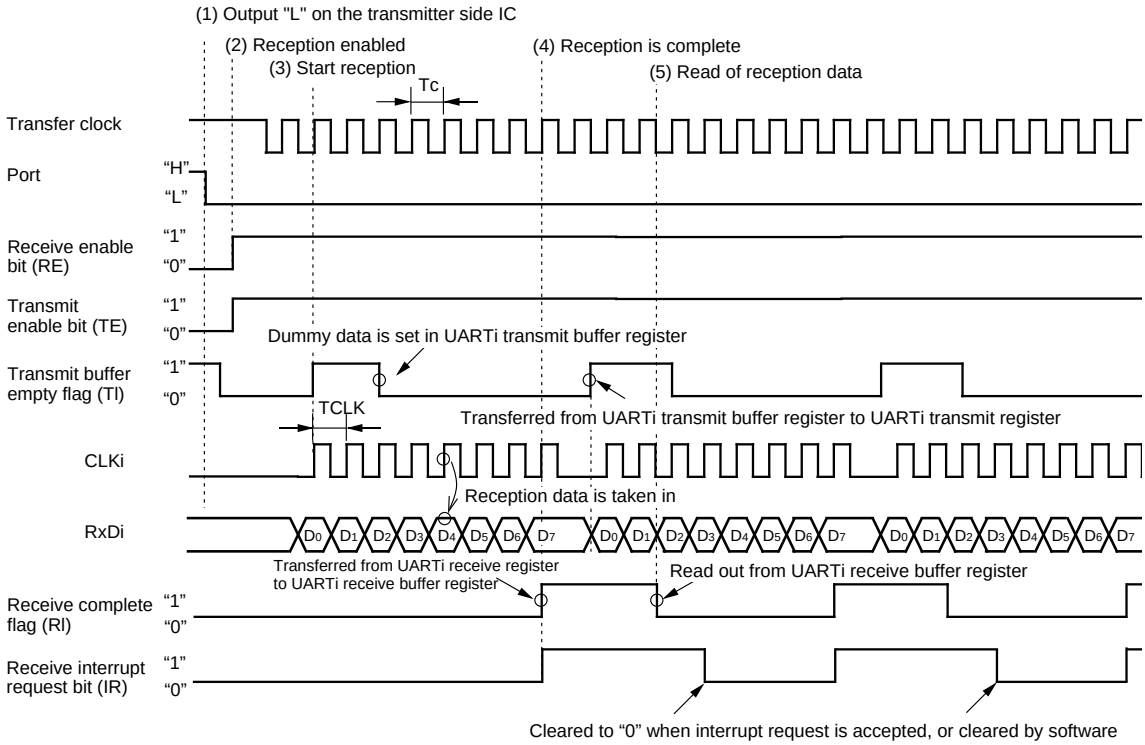
Operation
(1) Set an $\overline{SS}$ port of the transmitter side IC to output “L” level.
(2) Writing dummy data to the UARTi transmit buffer register, setting the receive enable bit to “1”, and the transmit enable bit to “1”, makes the data receivable status ready.
(3) In synchronization with the first rising edge of the transfer clock, the input signal to the RxDi pin is stored in the highest bit of the UARTi receive register. Then, data is taken in by shifting right the content of the UARTi reception data in synchronization with the rising edges of the transfer clock.
(4) When 1-byte data lines up in the UARTi receive register, the content of the UARTi receive register is transmitted to the UARTi receive buffer register. At this time, the receive complete flag and the UARTi receive interrupt request bit goes to “1”.
(5) The receive complete flag goes to “0” when the lower-order byte of the UARTi buffer register is read.

Note
• Set RxDi pins' port direction register to “0”.
• Set $\overline{SSi}$ pin to “H” level. If “L” level is input to the pin, a fault error will be generated.

Example of wiring



Example of operation


(1) Output "L" on the transmitter side IC
(2) Reception enabled
(3) Start reception
(4) Reception is complete
(5) Read of reception data
Tc
Transfer clock
Port
"H"
"L"
Receive enable bit (RE)
"1"
"0"
Transmit enable bit (TE)
Dummy data is set in UARTi transmit buffer register
Transmit buffer empty flag (TI)
TCLK
Transferred from UARTi transmit buffer register to UARTi transmit register
CLKi
Reception data is taken in
RxDi
D0 D1 D2 D3 D4 D5 D6 D7
Transferred from UARTi receive register to UARTi receive buffer register
Read out from UARTi receive buffer register
Receive complete flag (RI)
Receive interrupt request bit (IR)
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.
The above timing applies to the following settings:
- Internal clock is selected.
- CLK polarity select bit = "0".

$Tc = T_{CLK} = 2(n + 1) / fi$
fi: frequency of BRGi count source (f1, f8, f32)
n: value set to BRGi

Figure 2.5.11. Operation timing of reception in serial interface special function master mode, with clock delay

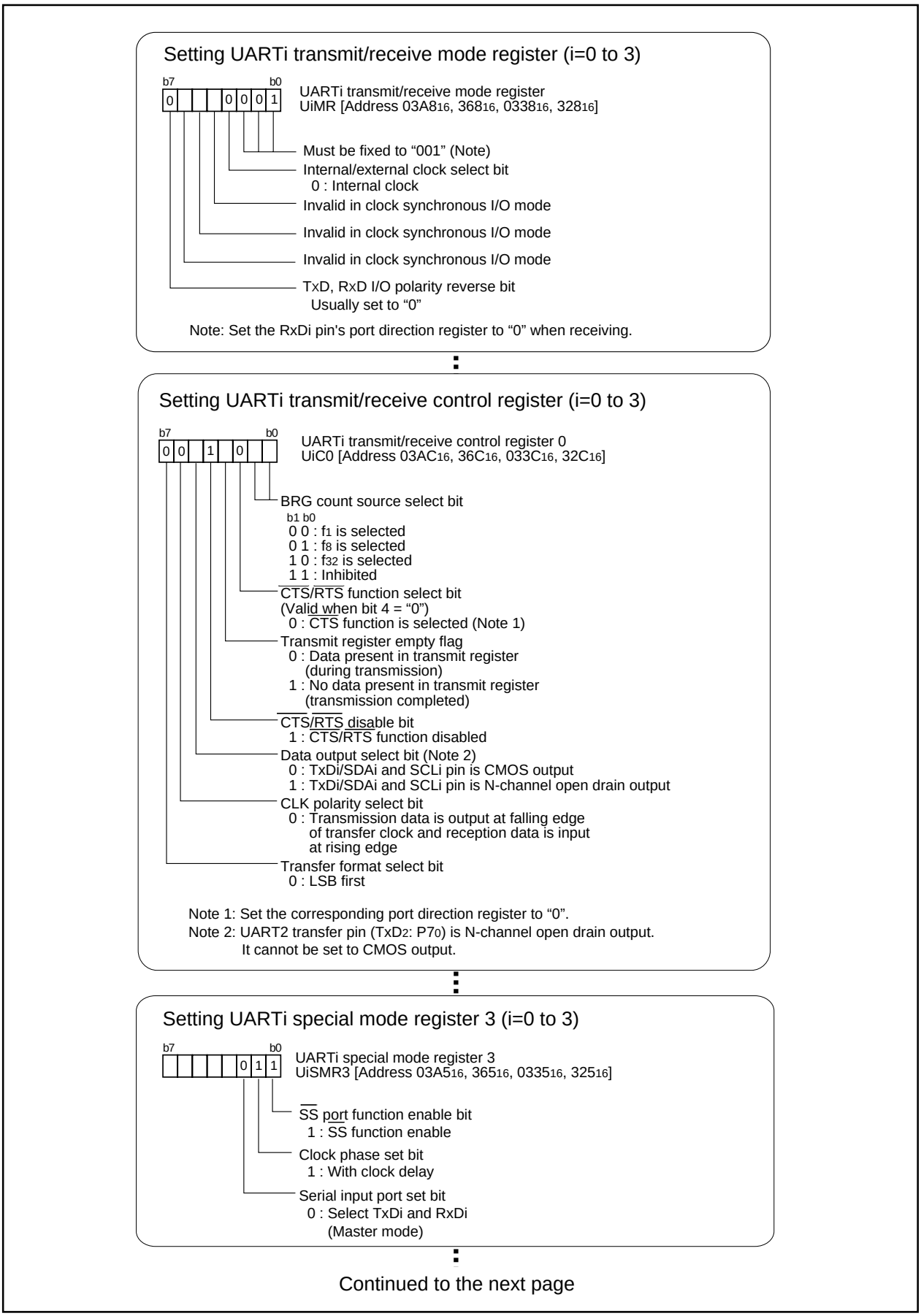

Setting UARTi transmit/receive mode register (i=0 to 3)
b7 b0
0 0 0 0 1
UARTi transmit/receive mode register
UiMR [Address 03A816, 36816, 033816, 32816]
Must be fixed to "001" (Note)
Internal/external clock select bit
0 : Internal clock
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
TxD, RxD I/O polarity reverse bit
Usually set to "0"
Note: Set the RxDi pin's port direction register to "0" when receiving.
Setting UARTi transmit/receive control register (i=0 to 3)
b7 b0
0 0 1 0
UARTi transmit/receive control register 0
UiC0 [Address 03AC16, 36C16, 033C16, 32C16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : Inhibited
CTS/RTS function select bit
(Valid when bit 4 = "0")
0 : CTS function is selected (Note 1)
Transmit register empty flag
0 : Data present in transmit register
(during transmission)
1 : No data present in transmit register
(transmission completed)
CTS/RTS disable bit
1 : CTS/RTS function disabled
Data output select bit (Note 2)
0 : TxDi/SDAi and SCLi pin is CMOS output
1 : TxDi/SDAi and SCLi pin is N-channel open drain output
CLK polarity select bit
0 : Transmission data is output at falling edge
of transfer clock and reception data is input
at rising edge
Transfer format select bit
0 : LSB first
Note 1: Set the corresponding port direction register to "0".
Note 2: UART2 transfer pin (TxD2: P70) is N-channel open drain output.
It cannot be set to CMOS output.
Setting UARTi special mode register 3 (i=0 to 3)
b7 b0
0 1 1
UARTi special mode register 3
UiSMR3 [Address 03A516, 36516, 033516, 32516]
SS port function enable bit
1 : SS function enable
Clock phase set bit
1 : With clock delay
Serial input port set bit
0 : Select TxDi and RxDi
(Master mode)
Continued to the next page

Figure 2.5.12. Set-up procedure of reception in serial interface special function master mode, with clock delay (1)

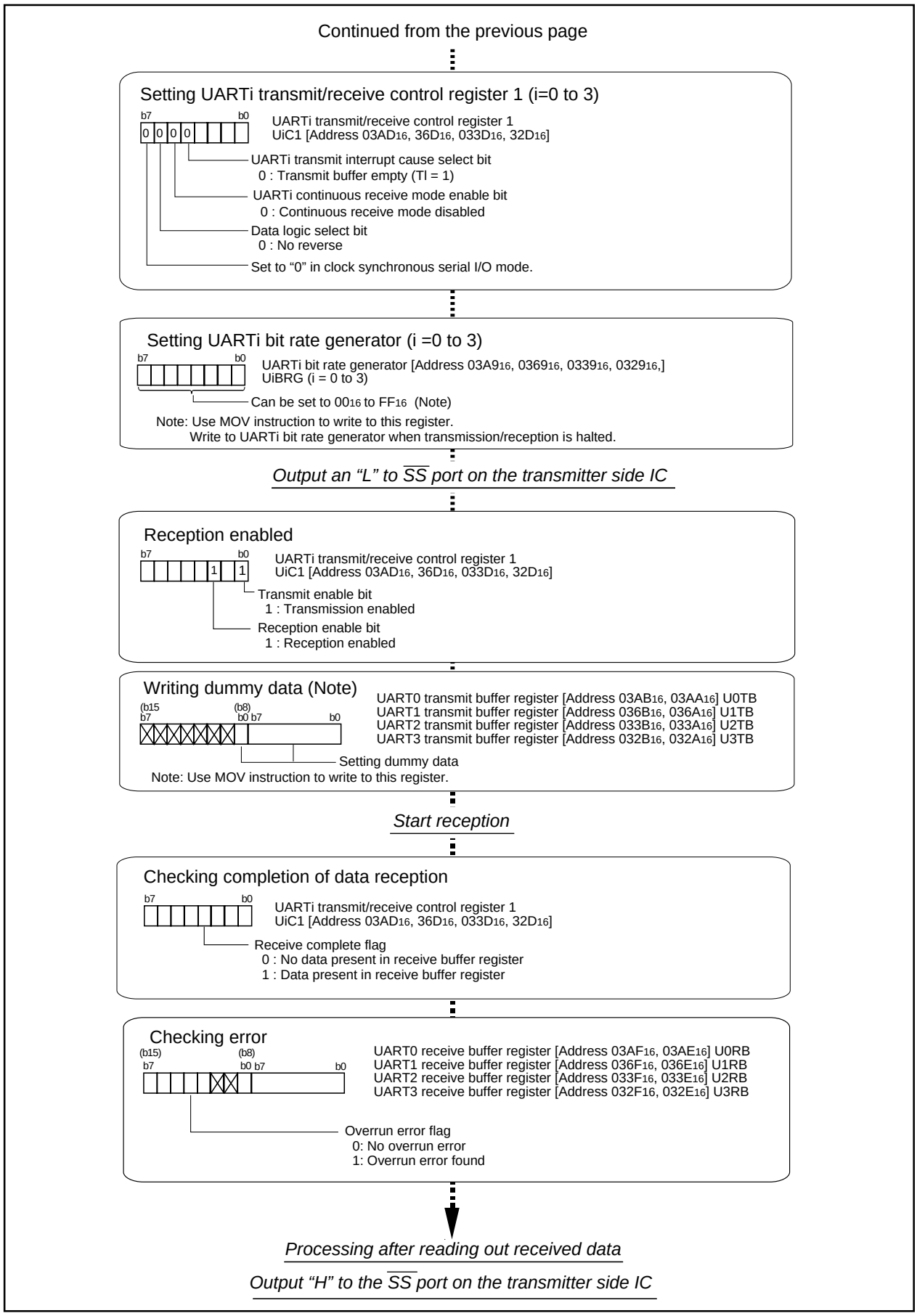

Continued from the previous page
Setting UARTi transmit/receive control register 1 (i=0 to 3)
b7 b0
0 0 0 0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
UARTi transmit interrupt cause select bit
0 : Transmit buffer empty (TI = 1)
UARTi continuous receive mode enable bit
0 : Continuous receive mode disabled
Data logic select bit
0 : No reverse
Set to "0" in clock synchronous serial I/O mode.
Setting UARTi bit rate generator (i =0 to 3)
b7 b0
UARTi bit rate generator [Address 03A916, 036916, 033916, 032916,]
UiBRG (i = 0 to 3)
Can be set to 0016 to FF16 (Note)
Note: Use MOV instruction to write to this register.
Write to UARTi bit rate generator when transmission/reception is halted.
Output an "L" to SS port on the transmitter side IC
Reception enabled
b7 b0
1 1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit enable bit
1 : Transmission enabled
Reception enable bit
1 : Reception enabled
Writing dummy data (Note)
(b15) (b8)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting dummy data
Note: Use MOV instruction to write to this register.
Start reception
Checking completion of data reception
b7 b0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Receive complete flag
0 : No data present in receive buffer register
1 : Data present in receive buffer register
Checking error
(b15) (b8)
b7 b0 b7 b0
UART0 receive buffer register [Address 03AF16, 03AE16] U0RB
UART1 receive buffer register [Address 036F16, 036E16] U1RB
UART2 receive buffer register [Address 033F16, 033E16] U2RB
UART3 receive buffer register [Address 032F16, 032E16] U3RB
Overrun error flag
0: No overrun error
1: Overrun error found
Processing after reading out received data
Output "H" to the SS port on the transmitter side IC

Figure 2.5.13. Set-up procedure of reception in serial interface special function master mode, with clock delay (2)

2.5.4 Operation of Serial Interface Special Function (transmission in slave mode without delay)

In transmitting data in serial interface special function slave mode, choose functions from those listed in Table 2.5.3. Operations of the circled items are described below. Figure 2.5.14 shows the operation timing, and Figures 2.5.15 and 2.5.16 show the set-up procedures.

Table 2.5.3. Choosed functions

Item	Set-up		Item	Set-up	
Transfer clock source		Internal clock (f_1 / f_8 / f_{32})	$\overline{SS}$i port function enable		$\overline{SS}$i function disabled
	O	External clock (CLKi pin)		O	$\overline{SS}$i function enabled
CLK polarity	O	Output transmission data at the falling edge of the transfer clock	Clock phase set	O	Without clock delay
					With clock delay
		Output transmission data at the rising edge of the transfer clock	Serial input port set		TxDi, RxDi selected (master mode)
Transmission interrupt factor	O	Transmission buffer empty		O	STxDi, SRxDi selected (slave mode)
		Transmission complete			

Operation (1) Input "L" level to an $\overline{SS}$i port by the output from the receiver side IC's port.

(2) Setting the transmit enable bit to “1” and writing transmission data to the UARTi transmit buffer register makes data transmissible status ready.

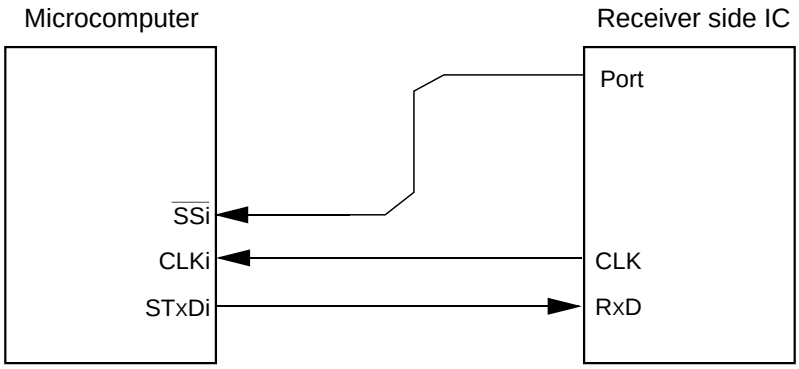
(3) In synchronization with the first falling edge of the transfer clock, transmission data held in the UARTi transmit buffer register is transmitted to the UARTi transmit register. At this time, the UARTi transmit interrupt request bit goes to “1”. Also, the first bit of the transmission data is transmitted from the STxDi pin. Then the data is transmitted bit by bit from the lower order in synchronization with the falling edges.

(4) When transmission of 1-byte data is completed, the transmit register empty flag goes to “1”, which indicates that transmission is completed.

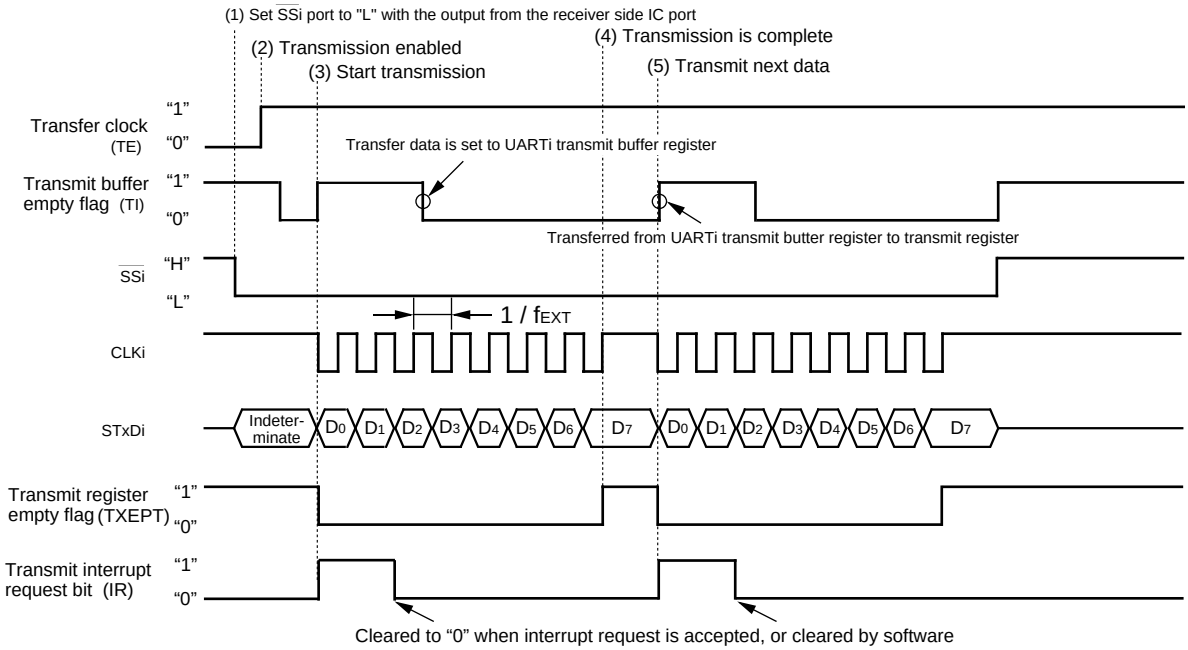
(5) If the next transmission data is set in the UARTi transmit buffer register while transmission is in progress (before the eighth bit has been transmitted), the data is transmitted in succession.

Note • Set CLKi pin's port direction register to “0”.

Example of wiring



Example of operation


(1) Set SSi port to "L" with the output from the receiver side IC port
(2) Transmission enabled
(3) Start transmission
(4) Transmission is complete
(5) Transmit next data
Transfer clock (TE)
Transfer data is set to UARTi transmit buffer register
Transmit buffer empty flag (TI)
Transferred from UARTi transmit butter register to transmit register
SSi
1 / fEXT
CLKi
STxDi
Indeter-minate
Transmit register empty flag (TXEPT)
Transmit interrupt request bit (IR)
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- External clock is selected.
- CLK polarity select bit = "0".

Make sure that the following conditions are met when the CLKi pin input ="H" before data reception
- Transmit enable bit →"1"
- Transmit data written to UARTi transmit buffer register

fEXT: frequency of external clock

Figure 2.5.14. Operation timing of transmission in serial interface special function slave mode, without clock delay

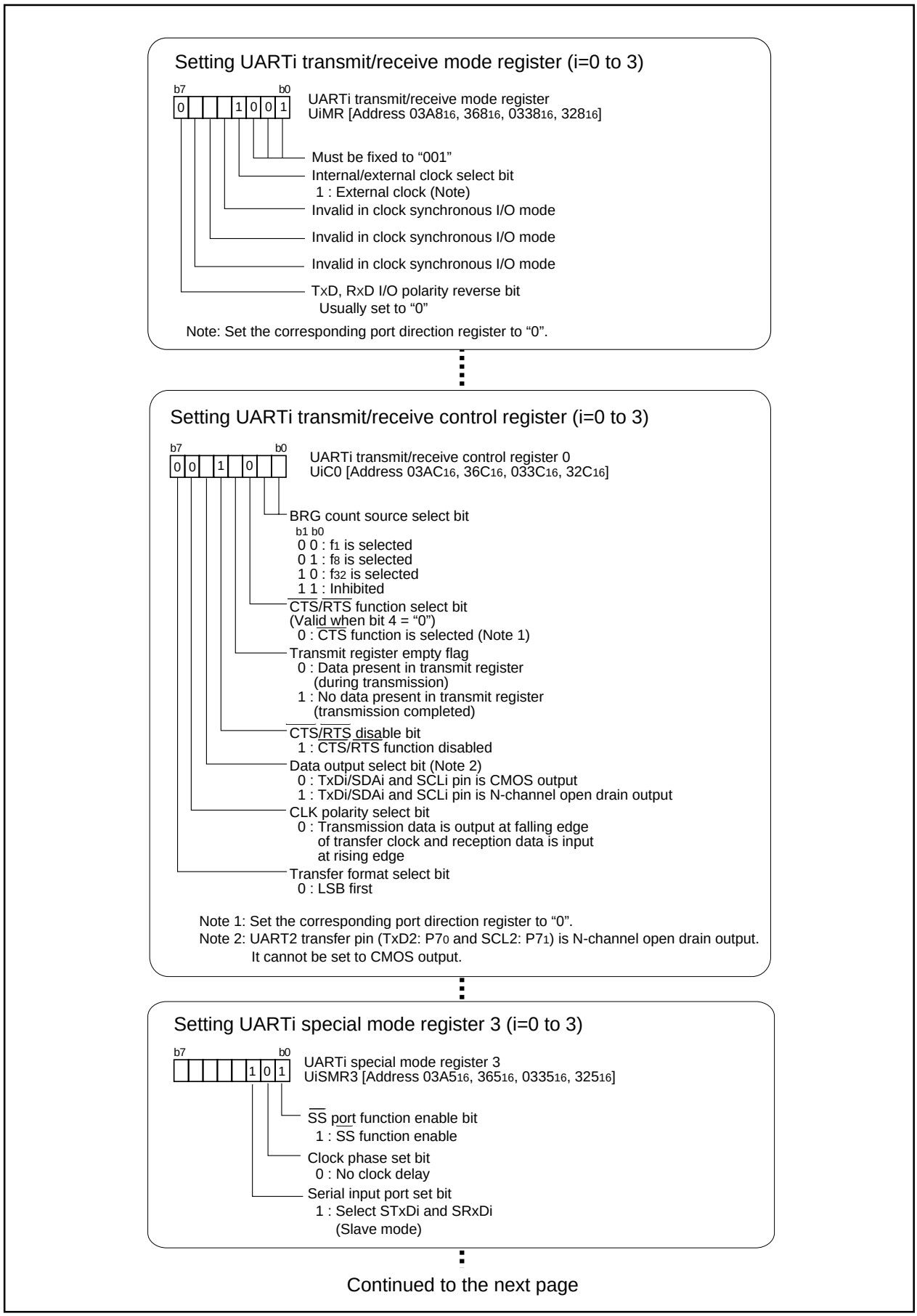

Setting UARTi transmit/receive mode register (i=0 to 3)
b7 b0
0 1 0 0 1
UARTi transmit/receive mode register
UiMR [Address 03A816, 36816, 033816, 32816]
Must be fixed to "001"
Internal/external clock select bit
1 : External clock (Note)
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
TXD, RXD I/O polarity reverse bit
Usually set to "0"
Note: Set the corresponding port direction register to "0".
Setting UARTi transmit/receive control register (i=0 to 3)
b7 b0
0 0 1 0
UARTi transmit/receive control register 0
UiC0 [Address 03AC16, 36C16, 033C16, 32C16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : Inhibited
CTS/RTS function select bit
(Valid when bit 4 = "0")
0 : CTS function is selected (Note 1)
Transmit register empty flag
0 : Data present in transmit register
(during transmission)
1 : No data present in transmit register
(transmission completed)
CTS/RTS disable bit
1 : CTS/RTS function disabled
Data output select bit (Note 2)
0 : TxDi/SDAi and SCLi pin is CMOS output
1 : TxDi/SDAi and SCLi pin is N-channel open drain output
CLK polarity select bit
0 : Transmission data is output at falling edge
of transfer clock and reception data is input
at rising edge
Transfer format select bit
0 : LSB first
Note 1: Set the corresponding port direction register to "0".
Note 2: UART2 transfer pin (TxD2: P70 and SCL2: P71) is N-channel open drain output.
It cannot be set to CMOS output.
Setting UARTi special mode register 3 (i=0 to 3)
b7 b0
1 0 1
UARTi special mode register 3
UiSMR3 [Address 03A516, 36516, 033516, 32516]
SS port function enable bit
1 : SS function enable
Clock phase set bit
0 : No clock delay
Serial input port set bit
1 : Select STxDi and SRxDi
(Slave mode)
Continued to the next page

Figure 2.5.15. Set-up procedure of transmission in serial interface special function slave mode, without clock delay (1)

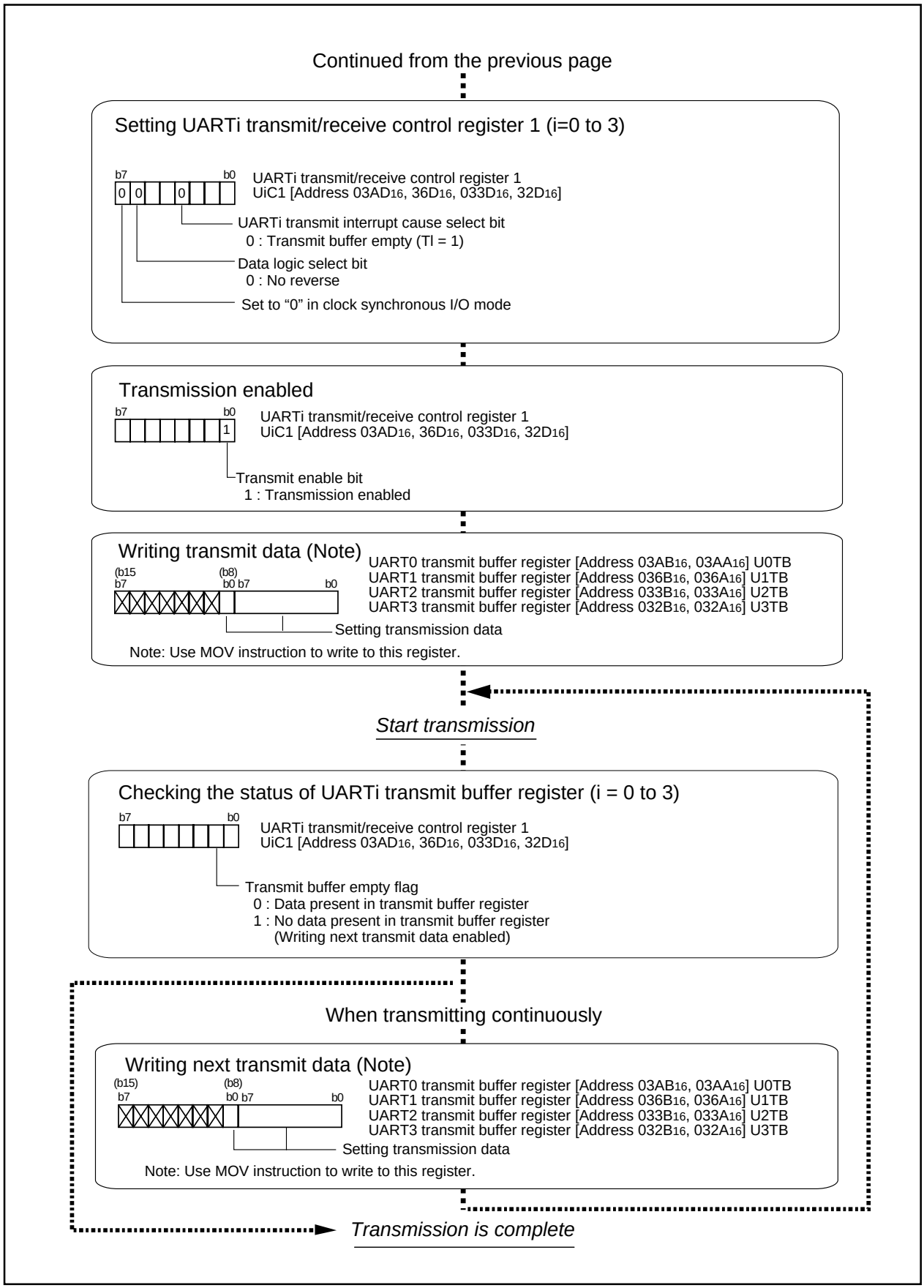

Continued from the previous page
Setting UARTi transmit/receive control register 1 (i=0 to 3)
b7 b0
0 0 0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
UARTi transmit interrupt cause select bit
0 : Transmit buffer empty (TI = 1)
Data logic select bit
0 : No reverse
Set to "0" in clock synchronous I/O mode
Transmission enabled
b7 b0
1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit enable bit
1 : Transmission enabled
Writing transmit data (Note)
(b15) (b8)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting transmission data
Note: Use MOV instruction to write to this register.
Start transmission
Checking the status of UARTi transmit buffer register (i = 0 to 3)
b7 b0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit buffer empty flag
0 : Data present in transmit buffer register
1 : No data present in transmit buffer register
(Writing next transmit data enabled)
When transmitting continuously
Writing next transmit data (Note)
(b15) (b8)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting transmission data
Note: Use MOV instruction to write to this register.
Transmission is complete

Figure 2.5.16. Set-up procedure of transmission in serial interface special function slave mode, without clock delay (2)

2.5.5 Operation of Serial Interface Special Function (reception in slave mode with clock delay)

In receiving data in serial interface special function slave mode, choose functions from those listed in Table 2.5.4. Operations of the circled items are described below. Figure 2.5.17 shows the operation timing, and Figures 2.5.18 and 2.5.19 show the set-up procedures.

Table 2.5.4. Choosed functions

Item	Set-up		Item	Set-up	
Transfer clock source		Internal clock (f_1 / f_8 / f_{32})	$\overline{SSi}$ port function enable		$\overline{SSi}$ function disabled
	O	External clock (CLKi pin)		O	$\overline{SSi}$ function enabled
CLK polarity	O	Output reception data at the rising edge of the transfer clock	Clock phase set		Without clock delay
				O	With clock delay
		Output reception data at the falling edge of the transfer clock	Serial input port set		TxDi, RxDi selected (master mode)
Continuous receive mode	O	Disabled		O	STxDi, SRxDi selected (slave mode)
		Enabled			

Operation (1) An $\overline{SSi}$ port is input "L" level which outputs from the transmitter side IC port.

(2) Writing dummy data to the UARTi transmit buffer register, setting the receive enable bit to “1”, and the transmit enable bit to “1”, makes the data receivable status ready.

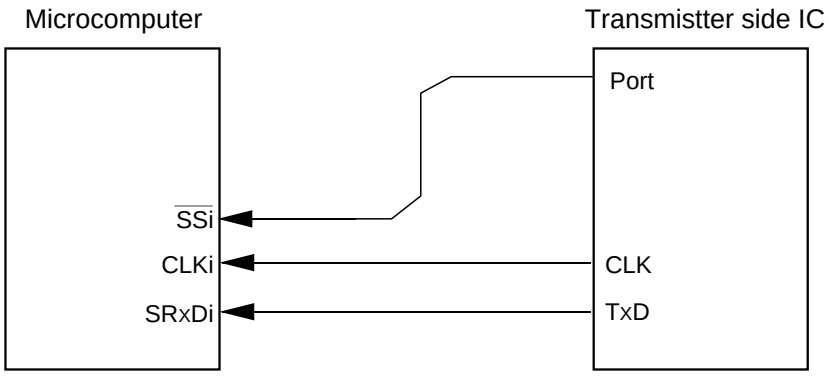
(3) In synchronization with the first rising edge of the transfer clock, the input signal to the SRxDi pin is stored in the highest bit of the UARTi receive register. Then, data is taken in by shifting right the content of the UARTi reception data in synchronization with the rising edges of the transfer clock.

(4) When 1-byte data lines up in the UARTi receive register, the content of the UARTi receive register is transmitted to the UARTi receive buffer register. At this time, the receive complete flag and the UARTi receive interrupt request bit goes to “1”.

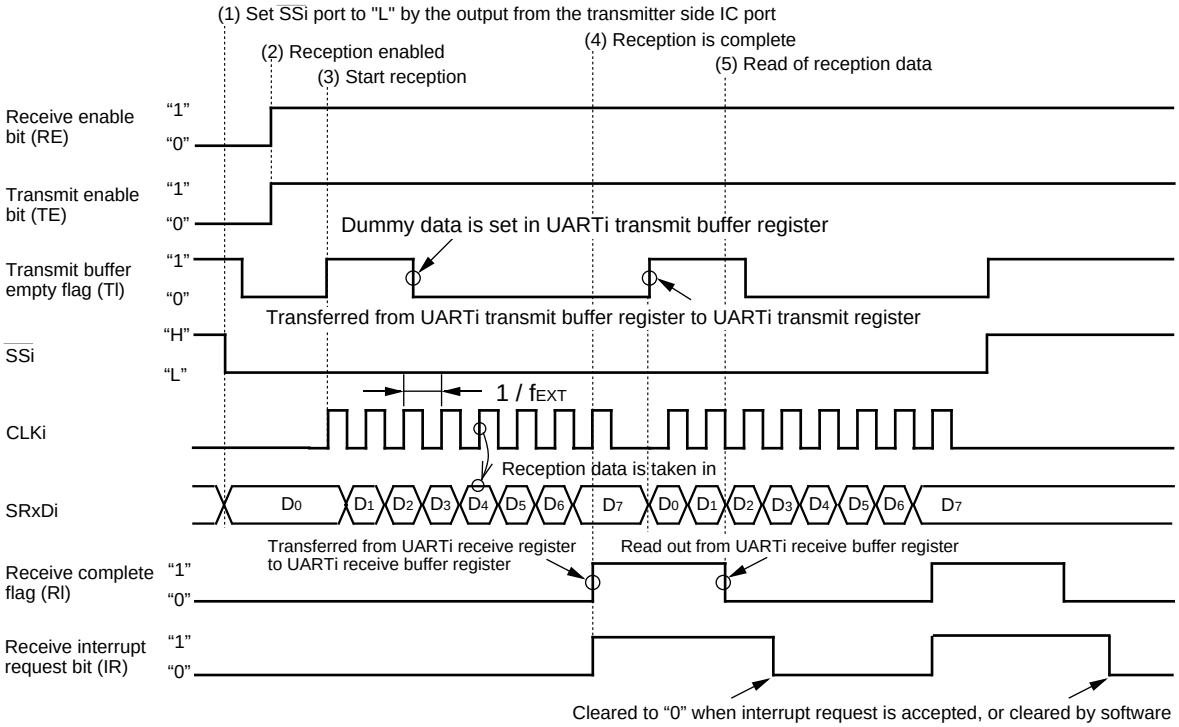
(5) The receive complete flag goes to “0” when the lower-order byte of the UARTi buffer register is read.

Note • Set CLKi and SRxDi pins' port direction register to “0”.

Example of wiring



Example of operation


(1) Set SSi port to "L" by the output from the transmitter side IC port
(2) Reception enabled
(3) Start reception
(4) Reception is complete
(5) Read of reception data
Receive enable bit (RE)
Transmit enable bit (TE)
Dummy data is set in UARTi transmit buffer register
Transmit buffer empty flag (TI)
Transferred from UARTi transmit buffer register to UARTi transmit register
SSi
1 / fEXT
CLKi
Reception data is taken in
SRxDi
Transferred from UARTi receive register to UARTi receive buffer register
Read out from UARTi receive buffer register
Receive complete flag (RI)
Receive interrupt request bit (IR)
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- External clock is selected.
- CLK polarity select bit = "0".

Make sure that the following conditions are met when the CLKi pin input ="H" before data reception
- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UARTi transmit buffer register

fEXT: frequency of external clock

Figure 2.5.17. Operation timing of reception in serial interface special function slave mode, with clock delay

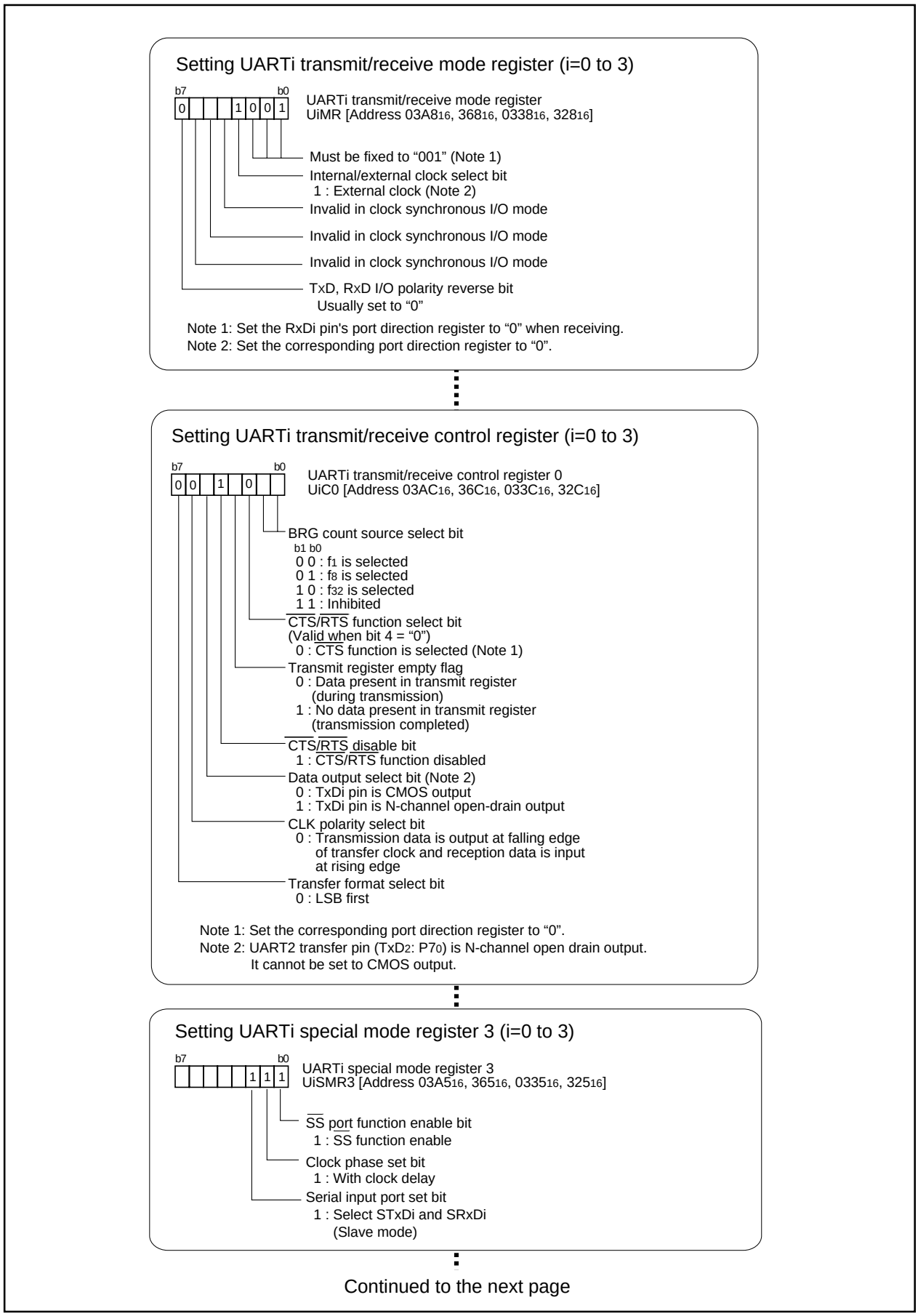

Setting UARTi transmit/receive mode register (i=0 to 3)
b7 b0
0 1 0 0 1
UARTi transmit/receive mode register
UiMR [Address 03A816, 36816, 033816, 32816]
Must be fixed to "001" (Note 1)
Internal/external clock select bit
1 : External clock (Note 2)
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
TxD, RxD I/O polarity reverse bit
Usually set to "0"
Note 1: Set the RxDi pin's port direction register to "0" when receiving.
Note 2: Set the corresponding port direction register to "0".
Setting UARTi transmit/receive control register (i=0 to 3)
b7 b0
0 0 1 0
UARTi transmit/receive control register 0
UiC0 [Address 03AC16, 36C16, 033C16, 32C16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : Inhibited
CTS/RTS function select bit
(Valid when bit 4 = "0")
0 : CTS function is selected (Note 1)
Transmit register empty flag
0 : Data present in transmit register (during transmission)
1 : No data present in transmit register (transmission completed)
CTS/RTS disable bit
1 : CTS/RTS function disabled
Data output select bit (Note 2)
0 : TxDi pin is CMOS output
1 : TxDi pin is N-channel open-drain output
CLK polarity select bit
0 : Transmission data is output at falling edge of transfer clock and reception data is input at rising edge
Transfer format select bit
0 : LSB first
Note 1: Set the corresponding port direction register to "0".
Note 2: UART2 transfer pin (TxD2: P70) is N-channel open drain output.
It cannot be set to CMOS output.
Setting UARTi special mode register 3 (i=0 to 3)
b7 b0
1 1 1
UARTi special mode register 3
UiSMR3 [Address 03A516, 36516, 033516, 32516]
SS port function enable bit
1 : SS function enable
Clock phase set bit
1 : With clock delay
Serial input port set bit
1 : Select STxDi and SRxDi (Slave mode)
Continued to the next page

Figure 2.5.18. Set-up procedure of reception in serial interface special function slave mode, with clock delay (1)

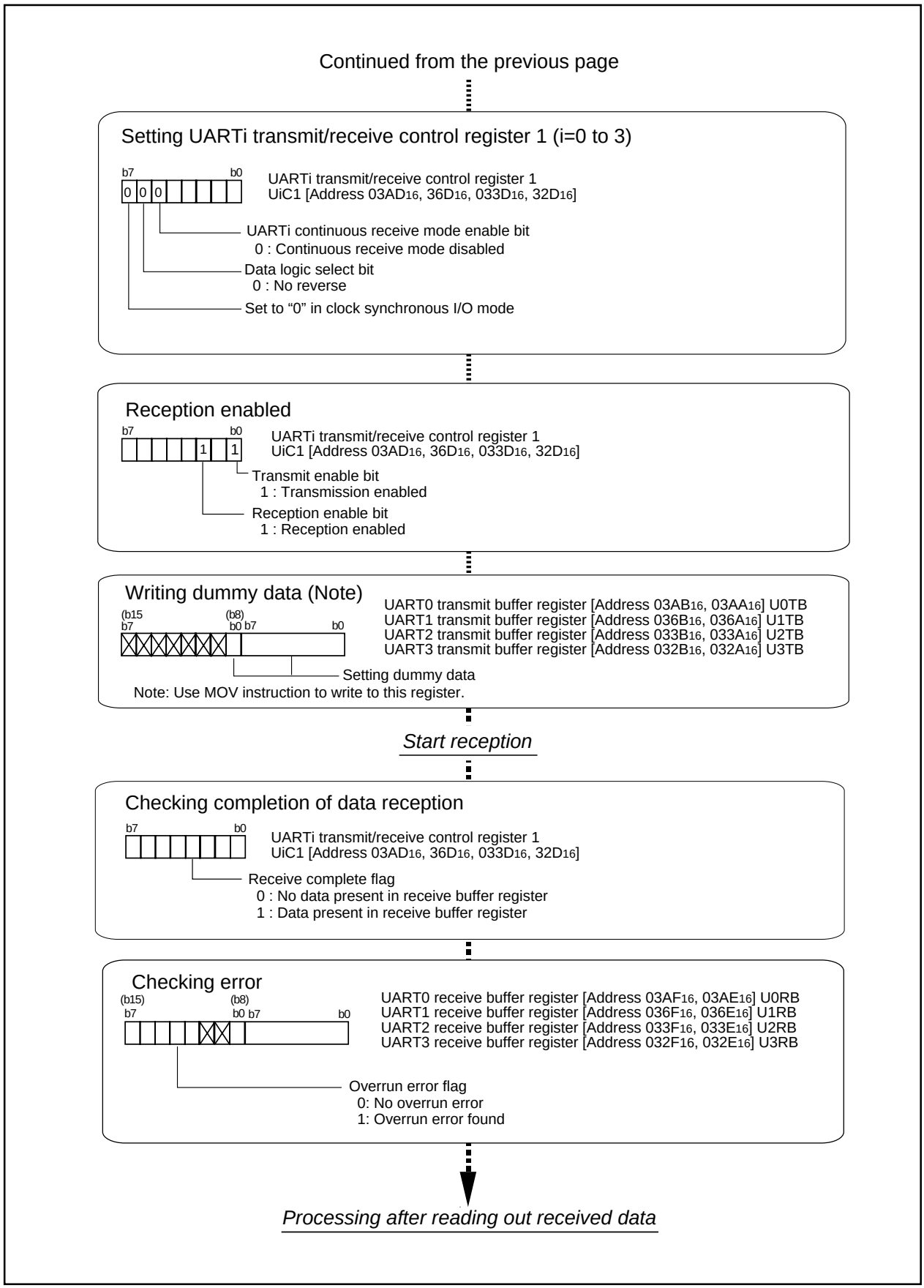

Continued from the previous page
Setting UARTi transmit/receive control register 1 (i=0 to 3)
b7 b0
0 0 0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
UARTi continuous receive mode enable bit
0 : Continuous receive mode disabled
Data logic select bit
0 : No reverse
Set to "0" in clock synchronous I/O mode
Reception enabled
b7 b0
1 1
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Transmit enable bit
1 : Transmission enabled
Reception enable bit
1 : Reception enabled
Writing dummy data (Note)
(b15) (b8)
b7 b0 b7 b0
UART0 transmit buffer register [Address 03AB16, 03AA16] U0TB
UART1 transmit buffer register [Address 036B16, 036A16] U1TB
UART2 transmit buffer register [Address 033B16, 033A16] U2TB
UART3 transmit buffer register [Address 032B16, 032A16] U3TB
Setting dummy data
Note: Use MOV instruction to write to this register.
Start reception
Checking completion of data reception
b7 b0
UARTi transmit/receive control register 1
UiC1 [Address 03AD16, 36D16, 033D16, 32D16]
Receive complete flag
0 : No data present in receive buffer register
1 : Data present in receive buffer register
Checking error
(b15) (b8)
b7 b0 b7 b0
UART0 receive buffer register [Address 03AF16, 03AE16] U0RB
UART1 receive buffer register [Address 036F16, 036E16] U1RB
UART2 receive buffer register [Address 033F16, 033E16] U2RB
UART3 receive buffer register [Address 032F16, 032E16] U3RB
Overrun error flag
0: No overrun error
1: Overrun error found
Processing after reading out received data

Figure 2.5.19. Set-up procedure of reception in serial interface special function slave mode, with clock delay (2)

2.6 Serial sound interface

2.6.1 Overview

The Serial Sound Interface (SSI) is a synchronous serial data interface used primary for transferring digital audio data. The bus of the 30245 Serial Sound Interface has four lines:

- Continuous serial clock (SCK)
- Word (channel) select (WS)
- Serial data out (XMIT)
- Serial data in (RX)

A basic Serial Sound Interface-based communication system has two Serial Sound Interfaces and a master controller, which generates both SCK and WS. The Serial Sound Interface that generates the control signals (SCK and WS) operates as a master, and the Serial Sound Interface that receives the external control signals operates as slave. The 30245 Sound Serial Interface can operate only as a slave. The transmitter/receiver must change channels on every WS transition. Through separate transmit and receive pins, simultaneous transmit and receive can be performed in synchronization with the same SCK and WS signals.

The following is an overview of the Serial Sound Interface.

● **Transmission/reception format**

The data path is designed to work with the data format of the USB audio class device specifications.

The transmitter/receiver must change channels on every WS transition. The number of SCKs within a WS high/low period is set as the channel width. The channel width can be selected from among 16 bits, 24 bits, and 32 bits using channel width select bits 0 and 1 (bits 4 and 5 of the SSIiMR0 register). If the number of SCKs exceeds the channel width, the receiver will stop receiving data until the next WS edge and the transmitter continues to transmit “0”. However, if the number of the SCKs falls short of the channel data width, both the transmitter and the receiver will immediately switch to transmit and receive, respectively, of the next channel data item.

● **Select function**

In the Serial Sound Interface function, the following features can be selected.

(1) Rate feedback function

When used with the USB interface, the Serial Sound Interface can count the number of WSs or SCKs per USB frame. The count value is loaded into the serial sound interface xRF register (x = 0 or 1) on the falling edge of each SOF pulse generated by the USB core. The SOF pulse is a frame delimiter used in USB communication. The value read from the register is the count from the immediately preceding USB frame.

(2) Channel width selection function

Channel widths of 32, 24 and 16 bits can be used to transmit and receive data. The width can be selected by the channel width select bits 0 and 1 in serial sound interface mode register 0.

(3) LSB/MSB first select function

This function is to choose whether to transmit/receive data from bit 1 or bit 7. This is valid when the transfer data length is 8 bits long.

Choose either of the following:

- LSB first Data is transmitted/received from bit 0.
- MSB first Data is transmitted/received from bit 7.

(4) Multiple receive format select function

If the number of SCKs in a WS high/low period is less than the channel width, the data can be placed either MSB or LSB justified.

(5) SCK polarity select function

This function selects whether transmit and receive data are synchronized to the rising or falling edge of WS. This is selected by the SCK polarity bit in the serial sound interface x mode register 1.

(6) WS polarity select function

This function is to transmit/receive data synchronized to the rising edge or the falling edge of WS. This is selected by the WS polarity select bit in the Serial Sound Interface x mode register 1.

(7) WS delay select function

Either of the following modes may be selected for the channel change timing:

- Normal WS mode
  WS transitions one SCK period before a channel change. (The channel changes one SCK period after WS transitions.)
- Delayed WS mode
  WS transitions concurrently with a channel change.

● **Input to the Serial Sound Interface function and the corresponding direction register**

When inputting an external signal to the Serial Sound Interface, set the corresponding port's direction register as input.

● **Serial Sound Interface function-related pins**

(1)	$SCLK_0$ and $SCLK_1$pins	Transfer clock input
(2)	WS_0 and WS_1pins	Channel clock input
(3)	RX_0 and RX_1pins	Data input
(4)	$XMIT_0$ and $XMIT_1$pins	Data output

● **Serial Sound Interface function-related register**

Figure 2.6.1 shows a memory map of the frequency synthesizer-related registers. Figures 2.6.2 and 2.6.3 show the configuration of Serial Sound Interface function related-registers, respectively.

Set the port's direction register appropriately and disable the clock synchronous serial and the UART which share the port.

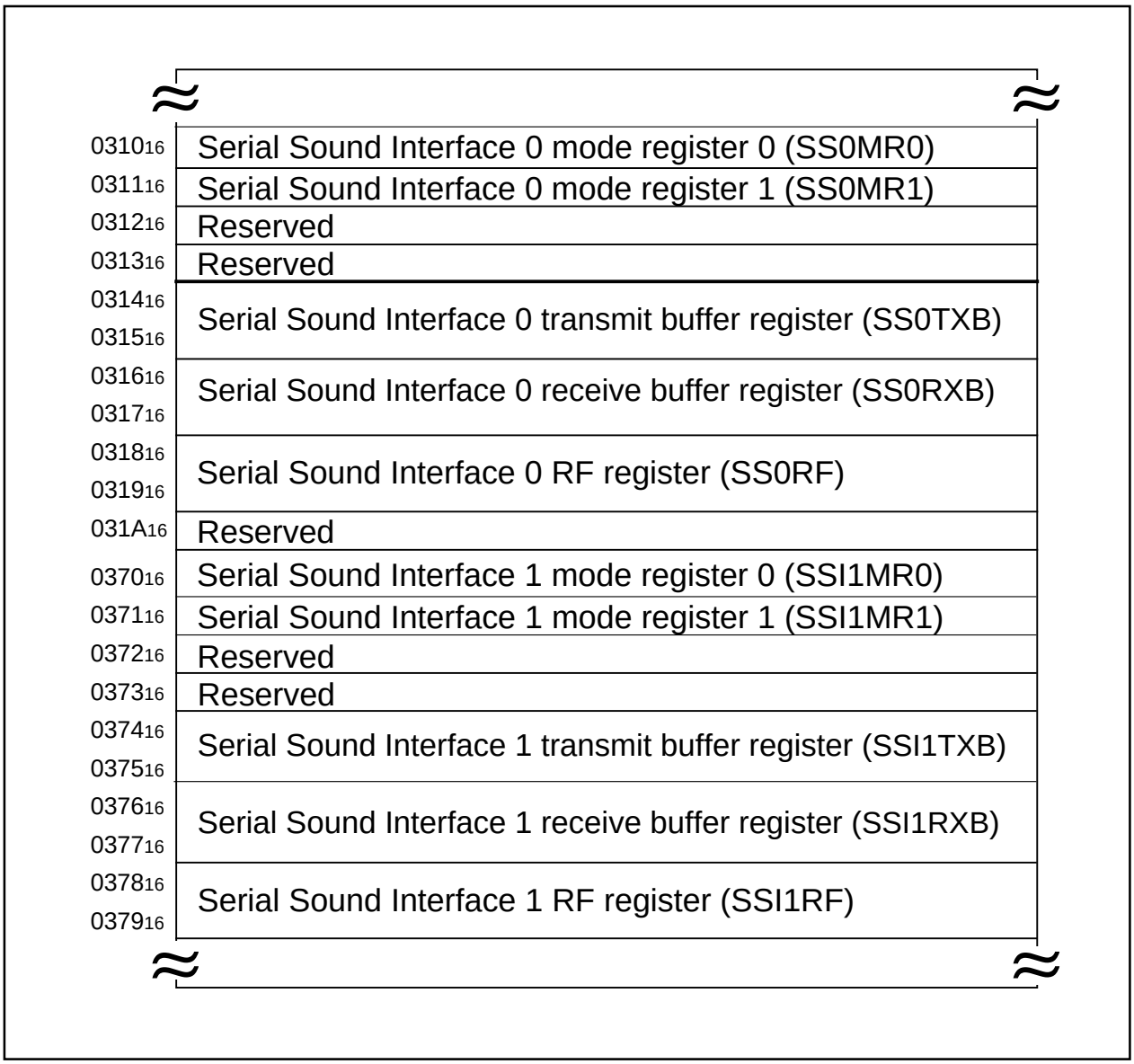
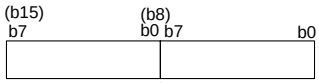


Figure 2.6.1. Memory map of Serial Sound Interface function-related registers

Serial Sound Interface x transmit buffer register

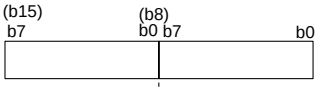


Symbol	Address	When reset
SSIxTXB (x = 0,1)	0315_{16}, 0314_{16} 0375_{16}, 0374_{16}	0000_{16}

Function	R	W
Transmit data (Note 1)	O	O

Note 1: For byte access, write data to addresses 0314_{16} and 0374_{16} only.
(Do not access to addresses 0315_{16} and 0375_{16}.)

Serial Sound Interface x receive buffer register

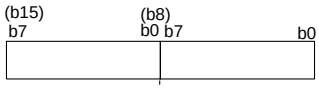


Symbol	Address	When reset
SSIxRXB (x = 0,1)	0317_{16}, 0316_{16} 0377_{16}, 0376_{16}	0000_{16}

Function	R	W
Receive data (Note 1)	O	O

Note 1: For byte access, write data to addresses 0316_{16} and 0376_{16}.
(Do not access to addresses 0317_{16} and 0377_{16}.)

Serial Sound Interface x RF register



Symbol	Address	When reset
SSIxRF (x= 0,1)	0319_{16}, 0318_{16} 0379_{16}, 0378_{16}	0000_{16}

Function	R	W
Rate feedback counter value	O	X

Figure 2.6.2. Serial Sound Interface function-related registers (1)

Serial Sound Interface x mode register 0

Symbol	Address	When reset
SSIxMR0 (x = 0,1)	0310_{16}, 0370_{16}	00_{16}

b7	b6	b5	b4	b3	b2	b1	b0

Bit symbol	Bit name	Function	R	W
SSIEN	Serial Sound Interface enable bit	0 : Disable 1 : Enable	O	O
XMTEN	Transmitter enable bit	0 : Disable 1 : Enable	O	O
RXEN	Receiver enable bit	0 : Disable 1 : Enable	O	O
RFBEN	Rate feedback counter enable bit	0 : Disable 1 : Enable	O	O
CWID0	Channel width select bit 0	b5 b4 0 0 : 32 bit 0 1 : 24 bit 1 0 : Disable 1 1 : 16 bit	O	O
CWID1	Channel width select bit 1		O	O
RFMT0	Receiver format select bit 0	0: LSB first 1: MSB first	O	O
RFMT1	Receiver format select bit 1	0 : LSB justified 1 : MSB justified	O	O

Serial Sound Interface x mode register 1

Symbol	Address	When reset
SSIxMR1 (x = 0,1)	0311_{16}, 0371_{16}	00_{16}

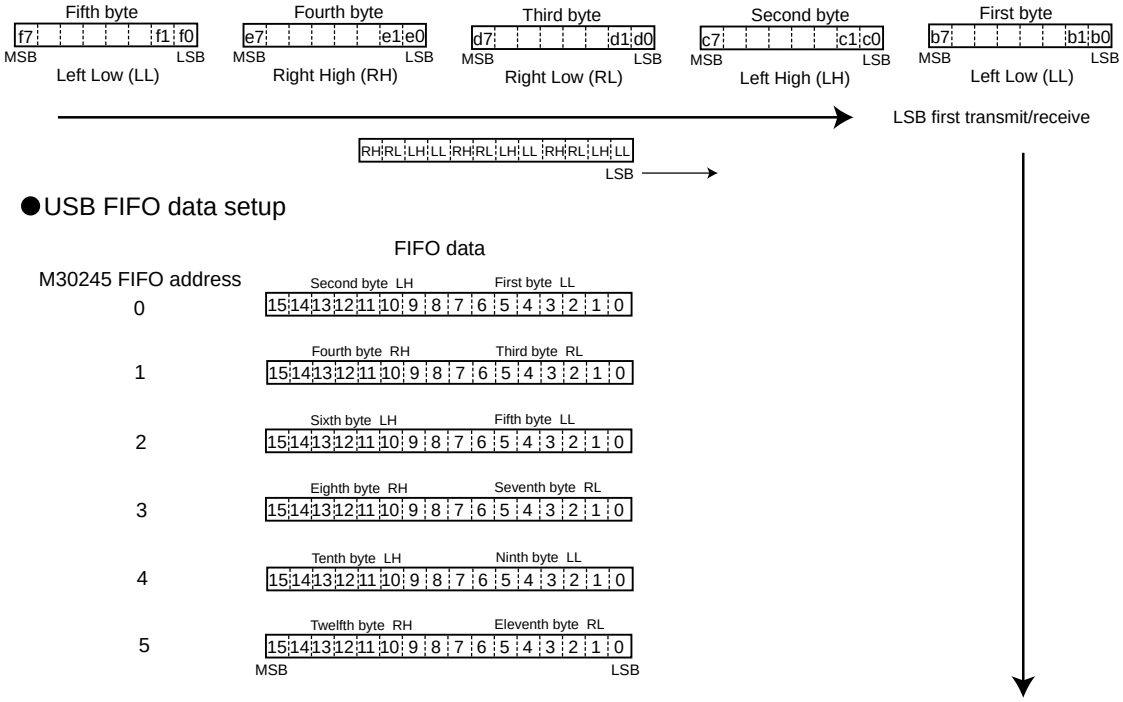
b7	b6	b5	b4	b3	b2	b1	b0
0	0					0	

Bit symbol	Bit name	Function	R	W
XMTFMT	Transmitter format	0: LSB first 1: MSB first	O	O
Reserved bit		Always set to “0”.	O	O
RFBSRC	Rate feedback counter source	0 : SCK 1 : WS	O	O
SCKP	SCK polarity select bit	0: Falling edge 1: Rising edge	O	O
WSP	WS polarity select bit	0: Falling edge 1: Rising edge	O	O
WSDLY	WS delay select bit	0: Delayed WS 1: Normal WS	O	O
Reserved bit		Always set to “0”.	O	O

Figure 2.6.3. Serial Sound Interface function-related registers (2)

In the case of the USB audio class, the following stream is output.

Audio stream (PCM) from the PC to the M30245 USB FIFO
For 16-bit data



●USB FIFO data setup

FIFO data

M30245 FIFO address	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0	Second byte LH / First byte LL
1	Fourth byte RH / Third byte RL
2	Sixth byte LH / Fifth byte LL
3	Eighth byte RH / Seventh byte RL
4	Tenth byte LH / Ninth byte LL
5	Twelfth byte RH / Eleventh byte RL

MSB LSB

○For 16-bit data

LSB b0 b7 b0 MSB LSB b7 b0 b7 b0 MSB b7

OPERATION	Left Buffer		Right Buffer	
	Byte 1	Byte 0	Byte 1	Byte 0
First Word Write	First byte	Second byte		
Second Word Write			Third byte	Fourth byte
Third Word Write	Fifth byte	Sixth byte		
Fourth Word Write			Seventh byte	Eighth byte

LL: b0 to b7 RL: b0 to b7
LH: b8 to b15 RH: b8 to b15

Just as 16 bits, the data width is expanded to output data for 24 bits and 32 bits .

○For 24-bit data

LSB b0 b7 b0 b7 b0 MSB LSB b7 b0 b7 b0 b7 b0 MSB b7

OPERATION	Left Buffer			Right Buffer		
	Byte 2	Byte 1	Byte 0	Byte 2	Byte 1	Byte 0
First Word Write	First byte (LL)	Second byte (LM)				
Second Word Write			Third byte (LH)	Fourth byte (RL)		
Third Word Write					Fifth byte (LM)	Sixth byte (LH)
Fourth Word Write	Seventh byte	Eighth byte				

LL: b0 to b7 RL: b0 to b7
LM: b8 to b15 RM: b8 to b15
LH: b16 to b23 RH: b16 to b23

○For 32-bit data

LSB b0 b7 b0 b7 b0 b7 b0 MSB LSB b7 b0 b7 b0 b7 b0 b7 b0 MSB b7

OPERATION	Left Buffer				Right Buffer			
	Byte 3	Byte 2	Byte 1	Byte 0	Byte 3	Byte 2	Byte 1	Byte 0
First Word Write	First byte (LL)	Second byte (LML)						
Second Word Write			Third byte (LMH)	Fourth byte (LH)				
Third Word Write					Fifth byte (RL)	Sixth byte (RML)		
Fourth Word Write							Seventh byte (RMH)	Eighth byte (RH)

LL: b0 to b7 RL: b0 to b7
LML: b8 to b15 RML: b8 to b15
LMH: b16 to b23 RHH: b16 to b23
LH: b24 to b31 RH: b24 to b31

Figure 2.6.4. Example of Audio stream (PCM) from the PC to the M30245 USB FIFO

2.6.2 Example of Serial Sound Interface operation

When using Serial Sound Interface (SSI), the DMA is recommended for reading and writing data quickly from the receive buffer to the transmit buffer. A programming example using DMA is shown below Figure 2.6.5 shows an example of Serial Sound Interface transmit timing, and Figure 2.6.6 shows an example of Serial Sound Interface receive timing.

```
/*****  Serial Sound Interface initialization routine ************************************************
        Serial Sound Interface is initialized by disabling.
        The DMA to use is also initialized.
        In this example, one DMA is used for each Serial Sound Interface input and output.
*************************************************************************************************************/
        ssi1mr1 = 0x00;                         /* SSI STOP  */
        ssi1mr0 = 0x00;                         /* SSI STOP  */
        dm0sl = 0x00;                           /* DMA0 STOP */
        dm0con = 0x00;                          /* DMA0 STOP */
        dm1sl = 0x00;                           /* DMA1 STOP */
        dm1con = 0x00;                          /* DMA1 STOP */

/*****  Setting example ******************************************************************************
        Audio transmission can be set before or after audio reception
        The DMA to use is set.
        DMA0 = audio data transmission and DMA1= audio data reception
*************************************************************************************************************/
        dm0ic  = 0x06;                          /* DMA completion interrupt enabled */
        dm0sl = 0x0e;                           /* DMA0 factor = SSI 1 transmit */
        sar0 = (unsigned long)&txb_buffer;      /* source address: buffer RAM, etc. */
        dar0 = (unsigned long)&ssi1txb;         /* destination address: SSI 1 transmit buffer */
        tcr0 = txb_counter;                     /* DMA0 transfer cycle setting */
        dm1ic = 0x06;                           /* DMA completion interrupt enabled */
        dm1sl = 0x0a;                           /* DMA1 factor = SSI 1 receive */
        sar1 = (unsigned long)&ssi1rxb;         /* source address: SSI 1 receive */
        dar1 = (unsigned long)&rxb_buffer;      /* destination address: buffer RAM, etc. */
        tcr1 = rxb_counter;                     /* DMA1 transfer cycle setting */

/*****  Activation processing routine ****************************************************************
        DMA activation. DMA0 can be activated before or after DMA1.
*************************************************************************************************************/
        dm0con = 0x18;                          /* DMA0 start [16bit, SRC = inc, single] */
        dm1con = 0x28;                          /* DMA1 start [16bit, DES = inc, single] */

/*******Serial Sound Interface is stopped at this point.
        Serial Sound Interface is enabled and audio data transmission/reception starts.***********/
```

(Continued to the next page.)

(Continued from the previous page.)

```
/*****  Serial Sound Interface activation routine for 16 bits ********************************************/
#ifdef OUT_Q_BIT_NO_16
        ssi1mr0 = 0x01;                         /* SSIEN = 1 */
        ssi1mr0 = 0xf1;                         /* 16bit / MSB justified */
        ssi1mr1 = 0x21;                         /* SCK neg, WS neg       MSB first normal  */
        ssi1mr0 = 0xf7;                         /* 16bit / Tx enable, Rx enable MSB justified */
#endif

/*****  Serial Sound Interface activation routine for 24 bits ********************************************/
#ifdef OUT_Q_BIT_NO_24
        ssi1mr0 = 0x01;                         /* SSIEN = 1 */
        ssi1mr0 = 0xd1;                         /* 24bit / MSB justified */
        ssi1mr1 = 0x21;                         /* SCK neg, WS neg       MSB first normal  */
        ssi1mr0 = 0xd7;                         /* 24bit / Tx enable, Rx enable MSB justified */
#endif

/*****  Serial Sound Interface activation routine for 32 bits ********************************************/
#ifdef OUT_Q_BIT_NO_32
        ssi1mr0 = 0x01;                         /* SSIEN = 1 */
        ssi1mr0 = 0xc1;                         /* 32bit / MSB justified */
        ssi1mr1 = 0x21;                         /* SCK neg, WS neg       MSB first normal  */
        ssi1mr0 = 0xc7;                         /* 32bit / Tx enable, Rx enable MSB justified */
#endif
```

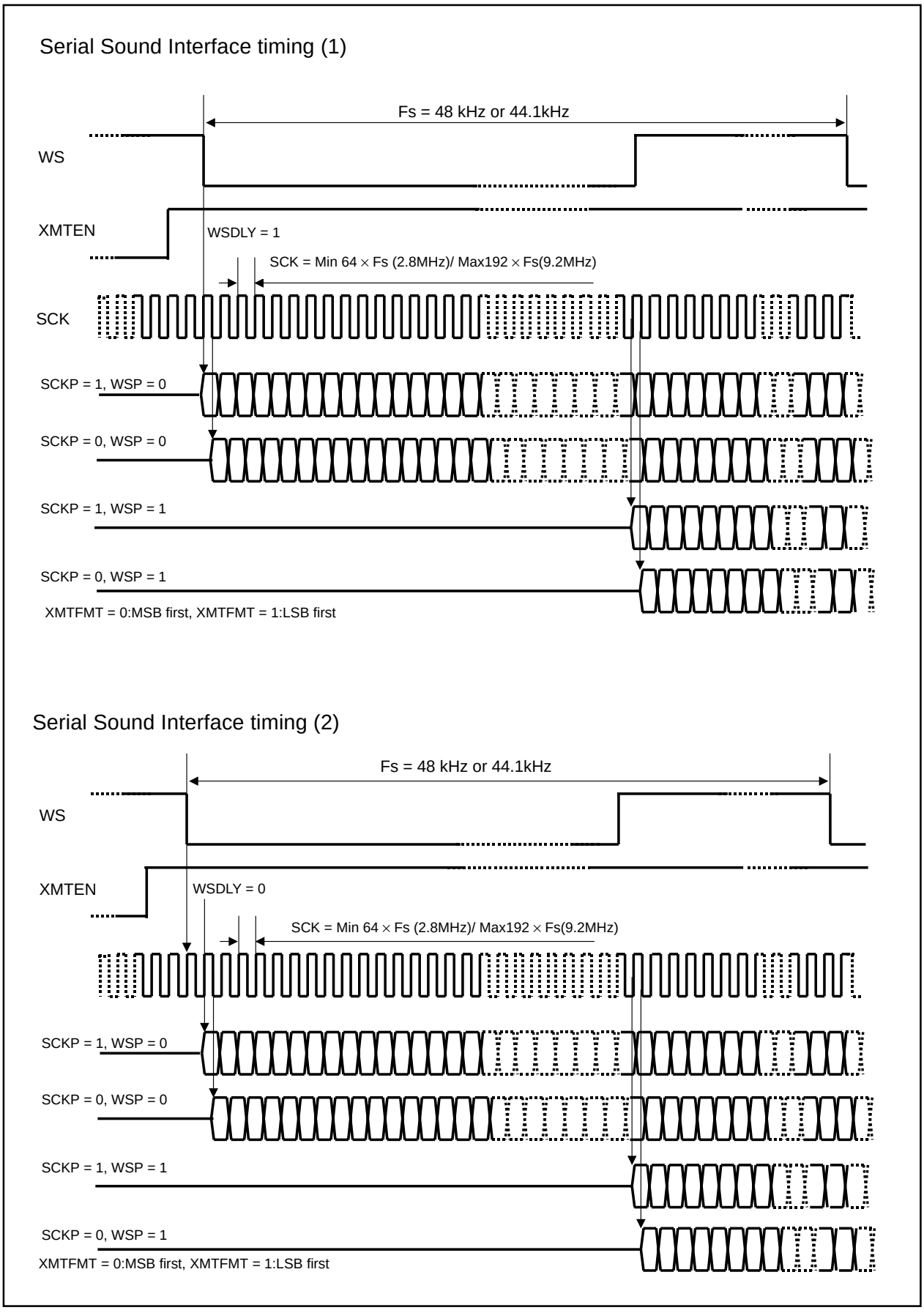

Serial Sound Interface timing (1)
Fs = 48 kHz or 44.1kHz
WS
XMTEN
WSDLY = 1
SCK = Min 64 × Fs (2.8MHz)/ Max192 × Fs(9.2MHz)
SCK
SCKP = 1, WSP = 0
SCKP = 0, WSP = 0
SCKP = 1, WSP = 1
SCKP = 0, WSP = 1
XMTFMT = 0:MSB first, XMTFMT = 1:LSB first
Serial Sound Interface timing (2)
Fs = 48 kHz or 44.1kHz
WS
XMTEN
WSDLY = 0
SCK = Min 64 × Fs (2.8MHz)/ Max192 × Fs(9.2MHz)
SCKP = 1, WSP = 0
SCKP = 0, WSP = 0
SCKP = 1, WSP = 1
SCKP = 0, WSP = 1
XMTFMT = 0:MSB first, XMTFMT = 1:LSB first

Figure 2.6.5. Example of Serial Sound Interface transmit timing

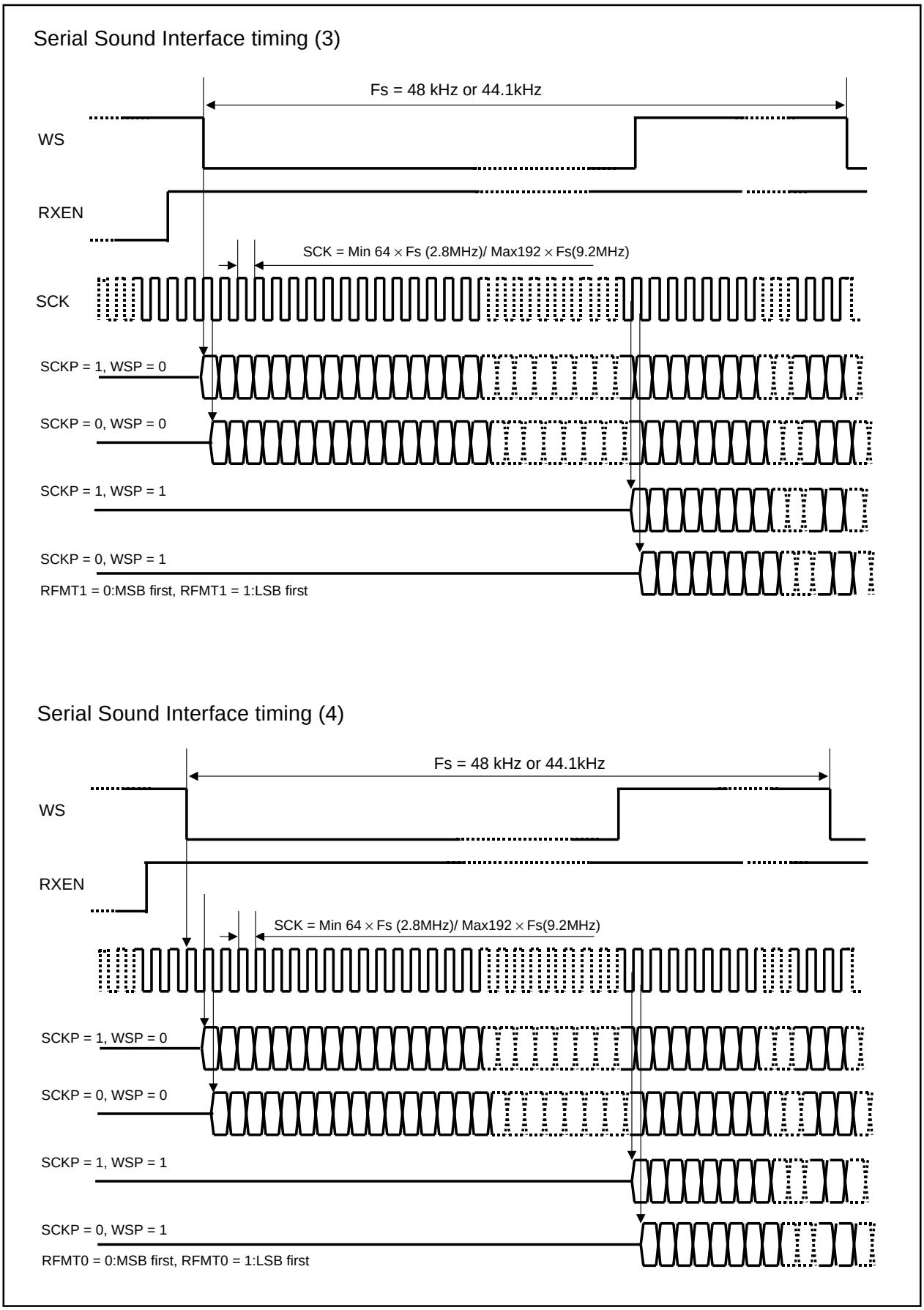

Serial Sound Interface timing (3)
Fs = 48 kHz or 44.1kHz
WS
RXEN
SCK = Min 64 × Fs (2.8MHz)/ Max192 × Fs(9.2MHz)
SCK
SCKP = 1, WSP = 0
SCKP = 0, WSP = 0
SCKP = 1, WSP = 1
SCKP = 0, WSP = 1
RFMT1 = 0:MSB first, RFMT1 = 1:LSB first
Serial Sound Interface timing (4)
Fs = 48 kHz or 44.1kHz
WS
RXEN
SCK = Min 64 × Fs (2.8MHz)/ Max192 × Fs(9.2MHz)
SCKP = 1, WSP = 0
SCKP = 0, WSP = 0
SCKP = 1, WSP = 1
SCKP = 0, WSP = 1
RFMT0 = 0:MSB first, RFMT0 = 1:LSB first

Figure 2.6.6. Example of Serial Sound Interface receive timing

2.6.3 Precautions for Serial Sound Interface

Description For flash memory version SSI transmission data must be latched as the following timing by a receiver.

- SCKP=0 (falling edge) : within 3 BCLK cycles from the rising edge of SCK
- SCKP=1 (rising edge) : within 3 BCLK cycles from the falling edge of SCK

2.7 Frequency synthesizer (PLL)

This paragraph explains the registers setting method and the notes related to the frequency synthesizer (PLL circuit).

2.7.1 Overview

The frequency synthesizer generates the 48MHz clock that is necessary for the USB block and the fSYN clock. These clocks are a multiple of the external input standard clock f(XIN). Figure 2.7.1 shows the frequency synthesizer circuit block diagram.

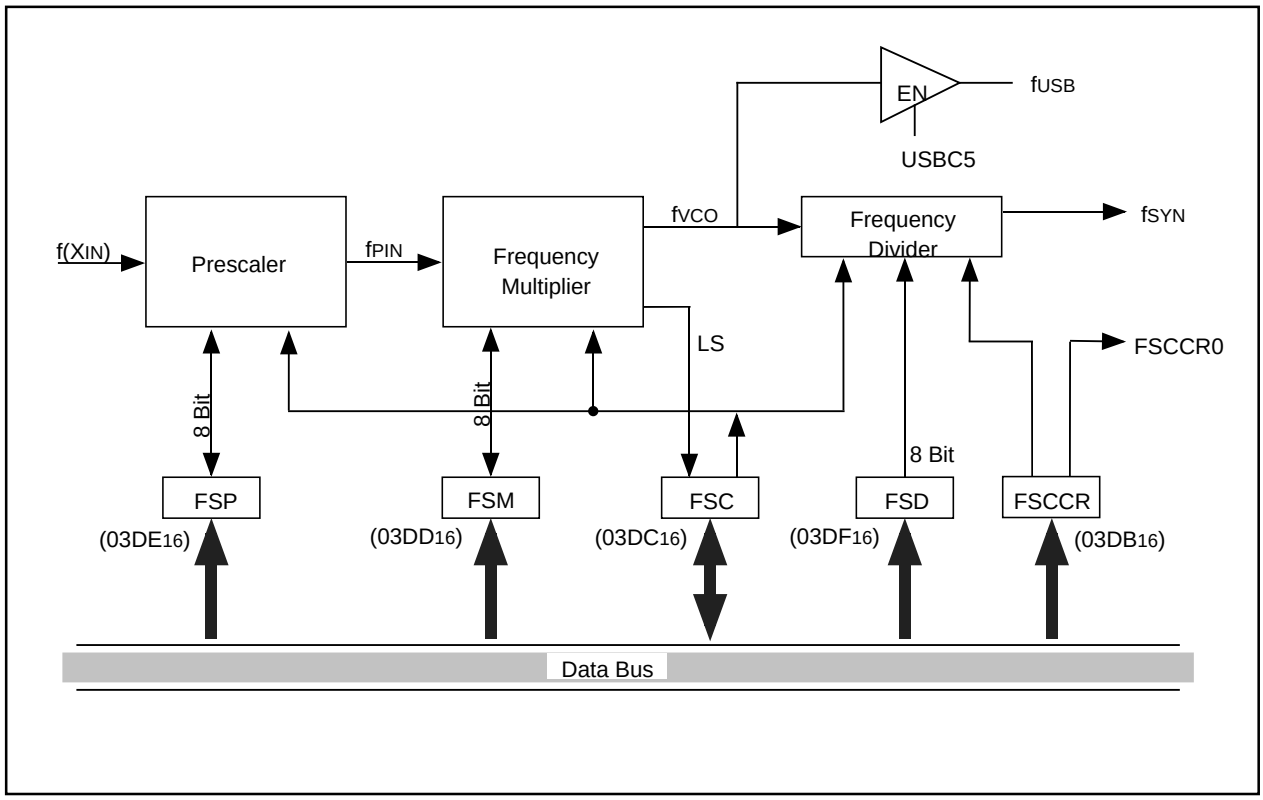


Figure 2.7.1. Frequency synthesizer circuit block diagram

(1) Related Registers

Figure 2.7.2 shows a memory location diagram for the frequency synthesizer related registers; Figures 2.7.3 and 2.7.4 show the composition of the frequency synthesizer related registers.

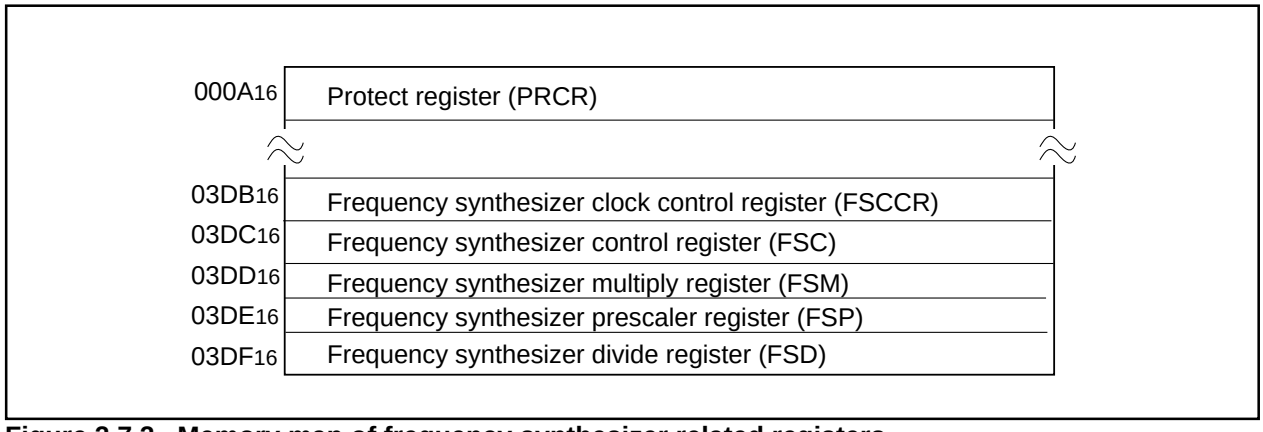


Figure 2.7.2. Memory map of frequency synthesizer related registers

Frequency Synthesizer Control register

b7	b6	b5	b4	b3	b2	b1	b0
			0	0			

Symbol	Address	When reset
FSC	$03DC_{16}$	01100000_2

Bit Symbol	Bit Name	Function	R	W
FSE	Frequency synthesizer enable bit	0 : Disabled 1 : Enabled	O	O
VCO0	VCO gain control bit	b2 b1 0 0 : Lowest gain 0 1 : Low gain 1 0 : High gain (Note 1) 1 1 : Highest gain	O	O
VC01			O	O
Reserved bit		Must always be set to “0”	O	O
CHG0	LPF current control bit (Note 2)	b6 b5 0 0 : Disabled 0 1 : Low current 1 0 : Medium current (Note 1) 1 1 : High current	O	O
CHG1			O	O
LS	Frequency synthesizer lock status bit	0 : Unlocked 1 : Locked	O	O

Note 1: Recommended.
Note 2: Bits 6 and 5 are set to (bit 6,bit 5)=(1,1) at reset. When using the frequency synthesizer, we recommend to set to (bit 6,bit 5)=(1,0).

Frequency Synthesizer Clock Control register

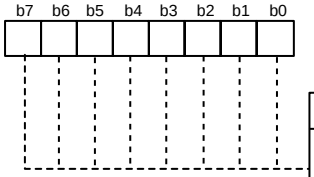
b7	b6	b5	b4	b3	b2	b1	b0
0	0	0		0	0	0	

Symbol	Address	When reset
FSCCR	$03DB_{16}$	00000000_2

Bit Symbol	Bit Name	Function	R	W
FSCCR0	Clock source selection	0 : Xin 1 : f_{SYN}	O	O
Reserved		Must always be set to “0”	O	O
FCCR4	Divide-by-3 option	0 : Normal 1 : Divide-by-3	O	O
Reserved		Must always be set to “0”	O	O

Figure 2.7.3. Frequency synthesizer registers (1)

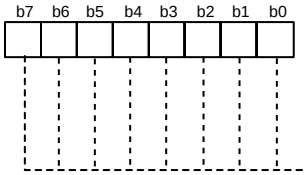
Frequency Synthesizer Prescaler Register



Symbol	Address	When reset
FSP	$03DE_{16}$	11111111_2

Bit Symbol	Bit Name	Function	R	W
FSP	Frequency synthesizer prescaler value	Generates f_{PIN} $f_{PIN} = f(X_{IN})/2(n + 1)$ n: FSP value	O	O

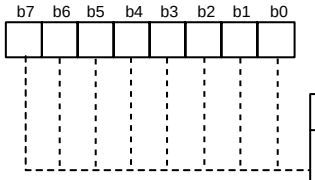
Frequency Synthesizer Multiply Register



Symbol	Address	When reset
FSM	$03DD_{16}$	11111111_2

Bit Symbol	Bit Name	Function	R	W
FSM	Frequency synthesizer multiplier value	Generates f_{VCO} by multiplying f_{PIN} $f_{VCO} = f_{PIN}$ X 2(n + 1) n: FSM value	O	O

Frequency Synthesizer Divide Register



Symbol	Address	When reset
FSD	$03DF_{16}$	11111111_2

Bit Symbol	Bit Name	Function	R	W
FSD	Frequency synthesizer divider value	Generates f_{SYN} by dividing f_{VCO} $f_{SYN} = f_{VCO}/2(m + 1)$ m: FSD value	O	O

Figure 2.7.4. Frequency synthesizer registers (2)

2.7.2 Operation of frequency synthesizer

The following explains how to setup after hardware reset. Table 2.7.1 to 2.7.3 show frequency synthesizer related registers setting examples.

Operation
(1) Cancel the protect register.
(2) Set the frequency synthesizer related registers to generate the 48MHz clock that is necessary for the f_{USB}.
(3) Enable the frequency synthesizer by setting frequency synthesizer control register.
(4) The protect register should be set to write disabled. A 3ms wait is necessary.
(5) The frequency synthesizer locked status bit should be checked. It is necessary to recheck after a wait of 0.1ms if it is “0” (unlocked).
(6) Enable USB clock.
(7) After waiting four cycles of the ϕ or greater, the USB enable bit should be set to “1”. A minimum delay of 250ns is needed before performing any other USB related registers read/write operations.

●Prescaler

Clock $f(X_{IN})$ is prescaled down by the frequency synthesizer prescaler register (FSP) to generate f_{PIN}. When the frequency synthesizer prescaler register is set at 255, the prescaler is disabled and $f_{PIN} = f(X_{IN})$. Table 2.7.1 shows some examples of how the frequency synthesizer prescaler register is set.

$f_{PIN} = f(X_{IN}) / 2(n+1)$ n: FSP value

Note: The value of f_{PIN} should not be set below 1 MHz.

Table 2.7.1. Example of setting the frequency synthesizer prescaler register (FSP)

f_{PIN}	FSP Value Dec (Hex)	$fX_{(IN)}$
12 MHz	255 (FF_{16})	12.00 MHz
1 MHz	7 (07_{16})	16.00 MHz
1 MHz	5 (05_{16})	12.00 MHz
2 MHz	3 (03_{16})	16.00 MHz
2 MHz	2 (02_{16})	12.00 MHz
3 MHz	1 (01_{16})	12.00 MHz
6 MHz	0 (00_{16})	12.00 MHz

●**Frequency Multiplier**

f_{VCO} is generated via the Frequency Synthesizer Mul-tiply register (FSM: address $03DD_{16}$). When the Frequency Multiply register is set to 255, multiplication is disabled and $f_{VCO} = f_{PIN}$. The value of n should be set so that f_{VCO} becomes 48MHz. Table 2.7.2 shows some examples of how the frequency synthesizer multiply register is set.

$f_{VCO} = f_{PIN} \times 2(n+1)$ n: FSM value

Table 2.7.2. Example of Setting the Frequency Multiply Register (FSM)

f_{PIN}	FSM Value Dec (Hex)	f_{VCO}
1 MHz	23 (17_{16})	48 MHz
2 MHz	11 ($0B_{16}$)	48 MHz
4 MHz	5 (05_{16})	48 MHz
6 MHz	3 (03_{16})	48 MHz
8 MHz	2 (02_{16})	48 MHz
12 MHz	1 (01_{16})	48 MHz

●**Frequency Divider**

Clock f_{SYN} is a divided down version of f_{VCO}. f_{SYN} is generated via the frequency synthesizer divide register (FSD). When the frequency synthesizer divider register is set to 255, division is disabled and $f_{SYN} = f_{VCO}$. Table 2.7.3 shows some examples of how the frequency synthesizer division register is set.

$f_{SYN} = f_{VCO} / 2(m+1)$ m: FSD value

Note 1: Set f_{SYN} to 12MHz or lower.

Table 2.7.3. Example of Setting the frequency synthesizer divide register (FSD)

f_{VCO}	FSD Value Dec (Hex)	f_{SYN}
48 MHz	1 (01_{16})	12.00 MHz
	2 (02_{16})	8.00 MHz
	2 (02_{16})	16.00 MHz (Note 1)
	3 (03_{16})	6.00 MHz
	127 ($7F_{16}$)	187.50 kHz

Note 1: $f_{SYN}=f_{VCO}/(m+1)$ when FSCCR4=1 and m=2.

2.7.3 Precautions for Frequency synthesizer

(1) Bits 6 and 5 of frequency synthesizer control register are set to (bit6, bit5)=(1, 1) at reset. When using the frequency synthesizer, we recommended to set to (bit6, bit5)=(1, 0).
(2) Set f_{SYN} to 12 MHz or lower.
(3) The value of f_{PIN} should not be set below 1 MHz.
(4) When the frequency synthesizer is enabled, do not use the output of the frequency synthesizer until after a 2~5ms delay. That will stabilize the output. Also, after the frequency synthesizer has been enabled, because the output is temporarily (2-5ms) unstable, the contents of none of the registers should be changed.
(5) When using the frequency synthesizer, connect a low pass filter to the LPF terminal.
(6) The following setup for the frequency synthesizer should be done after hardware resest.

1. Cancel the protect register.
2. Set the frequency synthesizer related registers to generate the 48MHz clock that is necessary for the f_{USB}.
3. Enable the frequency synthesizer by setting frequency synthesizer control register.
4. The protect register should be set to write disabled. A 3ms wait is necessary.
5. The frequency synthesizer locked status bit should be checked. It is necessary to recheck after a wait of 0.1ms if it is “0” (unlocked).
6. Enable USB clock.
7. After waiting four cycles of the f or greater, the USB enable bit should be set to “1”.
   A minimum delay of 250ns is needed before performing any other USB related registers read/write operations.

2.8 USB function

2.8.1 Overview

The USB function control unit of the M30245 group is compliant with USB2.0 specification and supports Full-Speed transfer. USB2.0 specification defines the following four kinds of transfer types:

- Control Transfer
- Isochronous Transfer
- Interrupt Transfer
- Bulk Transfer

The USB function control unit is provided with 9 endpoints including endpoint 0, endpoints 1 to 4 OUT (receive) and endpoints 1 to 4 IN (transmit), each of which having an FIFO. The endpoint 0 can apply only to the control transfer (same in transfer form as the bulk transfer); while endpoints 1 to 4 IN/OUT can apply to the bulk transfer, isochronous transfer and interrupt transfer. The size and the starting position of endpoints 1 to 4 IN/OUT FIFO can be set according to the user's system (The size and the starting position of endpoint 0 IN/OUT FIFO are fixed). Further, when the double buffer mode is enabled, the buffer which has twice as much as the set size is available for the IN/OUT FIFO. When the continuous receive/transmit mode is enabled, data can be transferred at a high speed (in bulk transfer).

The USB related interrupts include USB suspend interrupt, USB resume interrupt, USB reset interrupt, USB endpoint 0 interrupt, USB function interrupt, and USB SOF interrupt. The USB function control unit can control the state transition of the USB device based on these interrupt requests.

(1) Transfer Type

The USB specifications largely concern 2 types, including the one for the host side (PC/Hub) to control the connected peripheral devices and the other for the peripheral device side (Device) which is connected to the real machine. Further, depending on the number of data dealt with in the device, the peripheral devices which require more data (concerning image, audio, etc.) at one time have the communication specification with high transfer speed (12Mbps) called Full-Speed function, while the peripheral devices (keyboard, mouse) which require less data have the communication specification with low transfer speed (1.5Mbps) called Low-Speed function. Further, a communication specification with even higher speed is available which is called Hi-Speed function (480Mbps).

These communication specifications are each determined by device class of the peripheral devices and the transfer type to be used is determined for each peripheral device.

The M30245 group supports the following four kinds of transfer types:

- **Control Transfer**

  This transfer is used for communication of request-response form (bi-directional) in aperiodic communication which occurs suddenly. This is mainly used at the time of setup. As all the devices have to be supported in the standard device request, control transfer is supported without any exception for the devices compliant with USB.

- **Bulk Transfer**

  This transfer is used to transfer the data which delay does not cause any problem in aperiodic communication which occurs suddenly. This is used to transfer large quantities of data.

  Hardware detects errors in order to guarantee transfer data. A request for retransmission is issued by detecting of any error. For example, they include the output data from printer and image data of scanner.

● **Interrupt Transfer**

This transfer is used to notify the host of aperiodic and low-frequency data from the device. For example, they include the notification of out of paper in printer and data concerning devices such as the mouse and the keyboard.

● **Isochronous Transfer**

This transfer is used for continuous and periodic communication. Once the communication path is established, a transfer rate is guaranteed with a limited delay. The maximum size of transfer data is specified by the endpoint, which is read by the host as the configuration data of the device. Based on this data, transfer within the frame is scheduled and the bus time required for transfer of the maximum size of data is secured with preference. Although the band width and the transfer rate of data transfer are guaranteed, retransmission is not executed even if an error exists in the transfer. This is used for streaming data such as animation and audio data which requires real time.

(2) Communication Protocol

Host CPU has the initiative for the entire USB communication. Even when data are transmitted to the host from the device, the host gives the right of use to the device before data are transmitted. The host, in order to process multiple transfers simultaneously, schedules each transfer in packet unit within a frame of 1ms interval. The frames image is shown below:

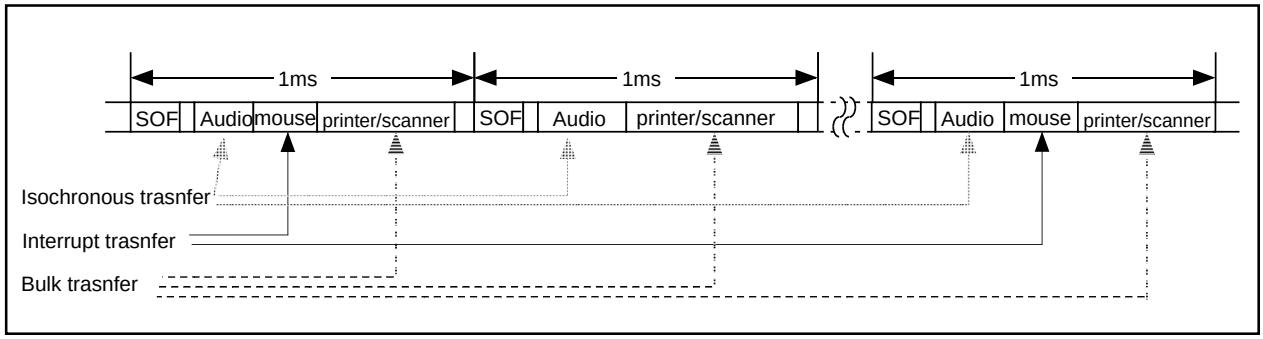


Figure 2.8.1. Frame image

● **Packet**

A packet is the unit in which the host CPU or the device secures a bus. In the USB, data are transmitted/received in the packet unit.

A packet is a group of bit data strings (fields), which starts with the SOP (Start-of-Packet) as a part of the SYNC (synchronous data) field. This is followed by the PID field which identifies a packet type and, then, each data field of a frame number/ address/ data field, etc., finally ending with EOP (End-of-Packet) which indicates the end of a packet. The packet types and formats are shown below:

SOF Packet: Packet to start the frame to be issued from the host for every 1ms.

8 bits	8 bits	5 bits
PID	Frame number	CRC5

PID: SOF(0xA5)

Token Packet: Packet to be issued from the host at the time of transaction start.

8 bits	7 bits	4 bits	5 bits
PID	ADDR	ENDP	CRC5

PID: OUT(0xE1),IN(0x69),SETUP(0x2D)

Data Packet: Packet to use at the time of data transfer.

8 bits	0 to 1034 bytes	16 bits
PID	DATA	CRC16

PID: DATA0(0xC3),DATA1(0x4B)

Hand Shake Packet: Packet to use at the transaction which controls flow.

8 bits
PID

PID: ACK(0xD2),NAK(0xA5),STALL(0x1E)

Note: In each packet, there are SOP as start of packet, and EOP as end of packet.

Figure 2.8.2. Kinds of packet

Table 2.8.1. List of USB packet recognitions

PID type	PID name	Process overview
Token	SETUP	Reports the operation request to device by the host CPU
	IN	Requests the data transmit to device by the host CPU
	OUT	Requests the data receive to device by the host CPU
	SOF	Indicates the start of frame to device by the host CPU
DATA	DATA0	Indicates that the sequence bit of transmit/receive data is even number
	DATA1	Indicates that the sequence bit of transmit/receive data is odd number
Handshake	ACK	Reports that the transmit data was correctly completed
	NAK	Reports that the device is in the communication wait state
	STALL	Reports that the communication was incorrectly completed

● **Transaction**

A transaction is the unit in which the host CPU schedules one frame. Each transaction is configured with packet, and the transaction types are determined according to the configuration pattern. The transaction types and formats are shown below:

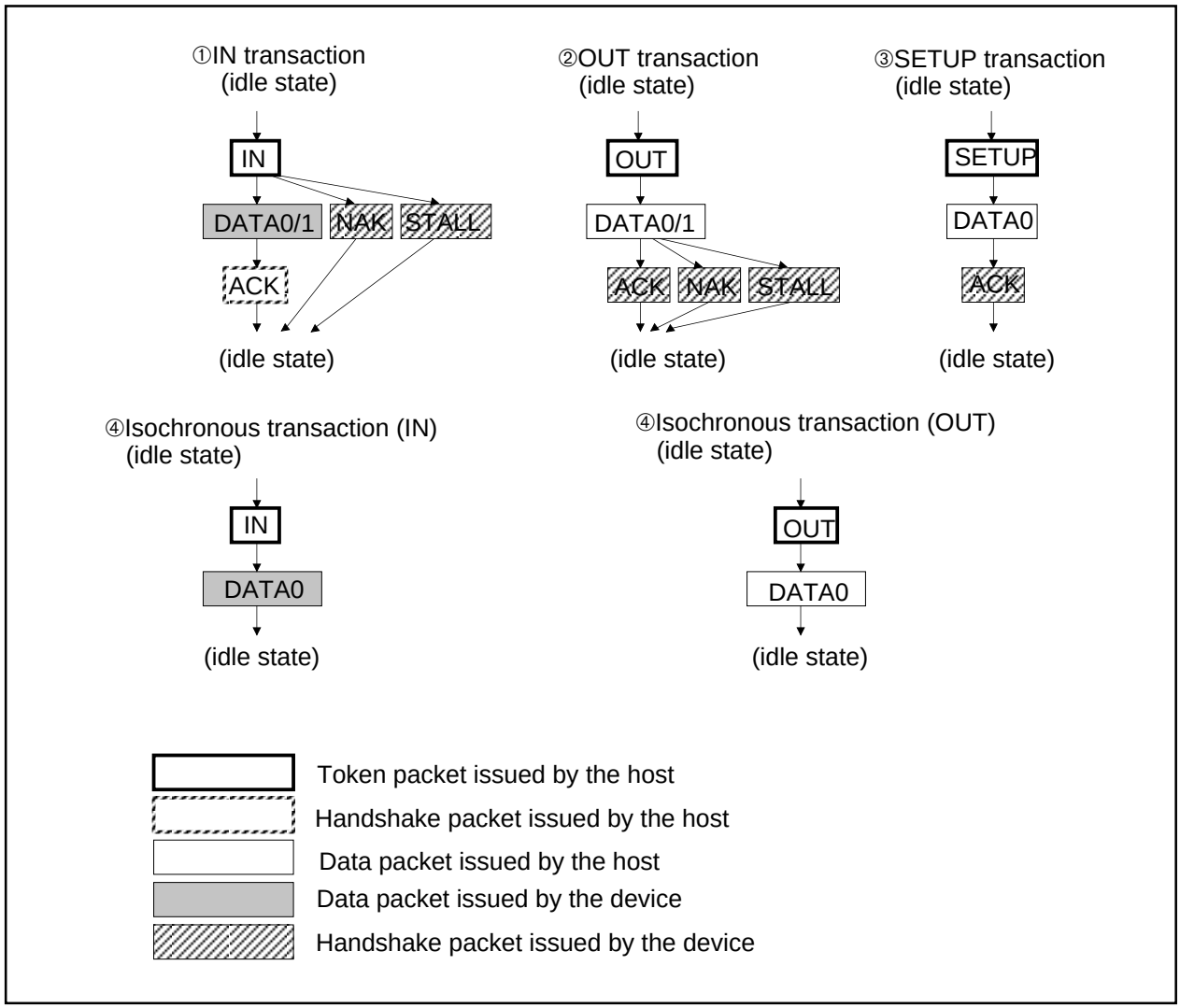


Figure 2.8.3. Formats of transactions

● **Communication Sequence**

The control transfer is used common to all devices at the time of setup, which consists of three kinds of stages being combined for one processing. The control transfer starts with setup stage. According to the content, data stage (control read transfer or control write transfer) is executed, followed by status stage being executed to finally complete one processing. In control transfer, use of endpoint 0 has been specified.

The communication sequence of control transfer is shown in Figure 2.8.4.

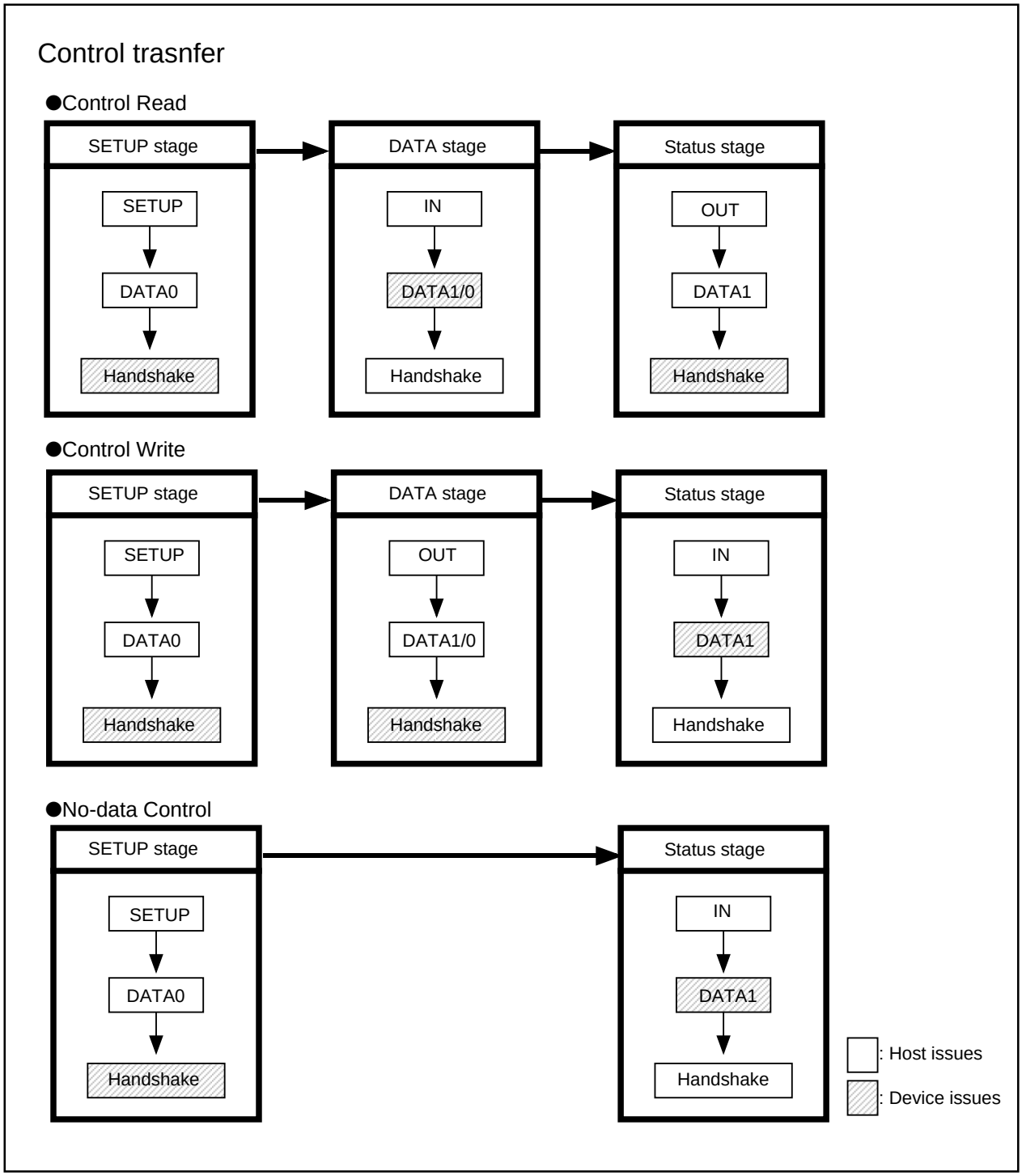


Figure 2.8.4. Control transfer communication sequence

● **Control Read Transfer**

In setup stage, host notifies the device that it is control read transfer. Then, in data stage, data are transmitted from the device to host through repetition of IN transaction. Finally, in status stage, OUT transaction is executed (that the host transmits an empty packet of data length (0) to the device) to complete the control read transfer.

● **Control Write Transfer**

In setup stage, host notifies the device that it is control write transfer. Then, in data stage, data are transmitted from the host to device through repetition of OUT transaction. Finally, in status stage, IN transaction is executed (that the device transmits an empty packet of data length (0) to the host) to complete the control write transfer.

● **No Control Data Transfer**

In setup stage, host notifies the device that it is no control data transfer. Then, in status stage, IN transaction is executed (that the device transmits an empty packet of data length (0) to the host) to complete the no control data transfer.

The execution result of setup stage and data stage are notified to host CPU in status stage. For details of response format of control transfers, refer to USB2.0 specification.

● **Device Request**

Concerning setup transaction in the setup stage of control transfer, format of its data phase has been defined, which is called device request.
For standard (0) type, it is called standard device request, which is the basic device request to be supported by all the USB devices.
For class (1) type, it is called class request. The USB implementers forum (USB IF) defines a device class, and determines the configuration required in the class and the class request.
For each data format of the device request, refer to USB2.0 specification or the specification for each class.

(3) Bulk Transfer

● Bulk IN Transfer

In bulk IN transfer which data are transmitted from the device to the host CPU, IN transactions are repeated. When transmit data are available in IN FIFO, the M30245 group issues a data packet to the IN token. When, during the handshake phase of each transaction, the M30245 group has normally received ACK packet issued by the host PC, it toggles DATA0 and DATA1 of data packet on next data phase. This serves to ensure handshake. The M30245 group executes the following responses when the data are not transmitted normally:

- •When the received IN token is destroyed, response is not executed.
- •When ACK handshake was not included in the transmit data, it is retransmitted on next IN token.
- •When the M30245 group was stalling, STALL handshake is returned.
- •When the transmit data are not available in IN FIFO, NAK handshake is returned.

● Bulk OUT Transfer

In bulk OUT transfer which data are transmitted from the host CPU to the device, OUT transactions are repeated.

The M30245 group has normally received a data packet, and then returns ACK handshake. Normal receiving is the status which is free of any bit stuffing error or CRC error and which data PID have been correctly received. When, during the handshake phase of each transaction, the host PC has normally received ACK packet issued by the M30245 group, it toggles DATA0 and DATA1 of data packet on next data phase. This serves to ensure handshake. The M30245 group executes the following responses when the data are not received normally:

- •When the received OUT token is destroyed, response is not executed.
- •When the M30245 group was stalling, STALL handshake is returned. Also, when the packet, which is exceeding receivable data size, is transmitted, STALL handshake is returned.
- •When inconsistency of the sequence bits is detected in the received data, ACK handshake is returned.
- •When OUT FIFO of the M30245 group could not receive full data, NAK handshake is returned.

For details, refer to USB2.0 specification.

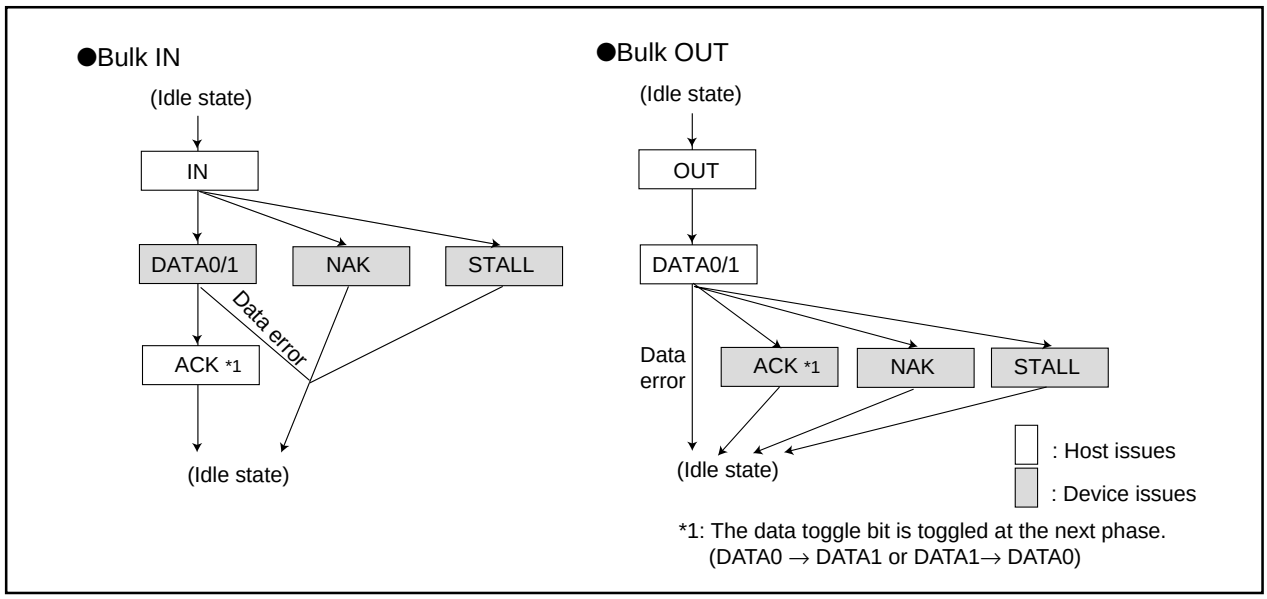


Figure 2.8.5. Bulk transfer

(4) Isochronous Transfer

● Isochronous IN Transfer

In isochronous IN transfer which data are transferred from the device to the host CPU, isochronous (IN) transactions are repeated. Isochronous transaction does not have the handshake phase. The data packet consists only of DATA0. Toggling with DATA1 is not performed. When transmit data is available in IN FIFO, the M30245 group issues a data packet to the IN token. The M30245 group executes the following responses when the data are not transmitted normally:

- •When the received IN token is destroyed, the data are not issued.
- •When the transmit data are not available in IN FIFO, an empty packet of data length (0) is issued.

● Isochronous OUT Transfer

In isochronous OUT transfer which data are transferred from the host CPU to the device, isochronous (OUT) transactions are repeated. Isochronous transaction does not have the handshake phase. The data packet consists only of DATA0. Toggling with DATA1 is not performed.

The M30245 group has received a data packet, indicates whether or not the data content is normal by using a status flag. The M30245 group executes the following responses when the data are not received normally:

- •When the received OUT token is destroyed, the data are not received.
- •When the received data are destroyed (bit stuffing error or CRC error occur), the data are received.
- •When the packet, which is exceeding receivable data size, is transmitted, the data are not received.
- •When OUT FIFO of the M30245 group could not receive the full data, the data are not received.

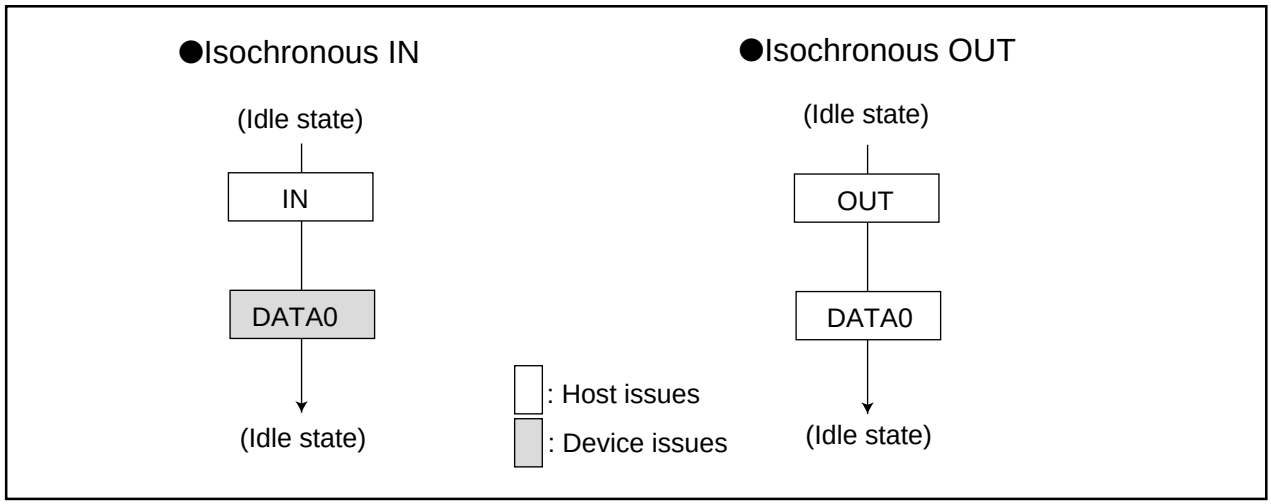


Figure 2.8.6. Isochronous transfer

(5) Interrupt Transfer

This transfer form is same as the bulk transfer. Refer to “(3) Bulk Transfer” of this manual.

(6) Device State

The device has states and, transits between the states. The M30245 group does not execute state transition on the hardware. Control it by the software based on requests of the related USB interrupt request. The device state transition is shown in Figure 2.8.7.

A series of processes from bus connection to configuration is called “emulation”. Each state is explained as follows:

①Connection State:

This is the state which the device has been connected to the bus.

②Powered State:

This is the state where the hub has been completed the configuration and has been supplied power to the bus.

③Default State:

This is the state where the reset signal has been received from the host CPU. It is responded as default address (0). This is the unconfigured state (configuration 0).

④Address State:

This is the state which the SET_ADDRESS standard device request has been received and a device address other than “0” has been assigned. This is the unconfigured state (Configuration 0).

⑤Configured State

This is the state which endpoint 0 has been received the SET_CONFIGURATION standard device request and the device has been configured.

⑥Suspend State

This is the state which inactive state followed 3ms or more.

If a bus active has been detected, the state shifts to the former one. In the M30245 group, when a bus reset is detected in suspend state, detects bus active and shifts to the former state, then, detects the bus reset.

(7) USB Related Registers Memory Mapping

The USB related registers memory mapping is shown in Figure 2.8.8. The list of USB related registers items is shown in Table 2.8.2.

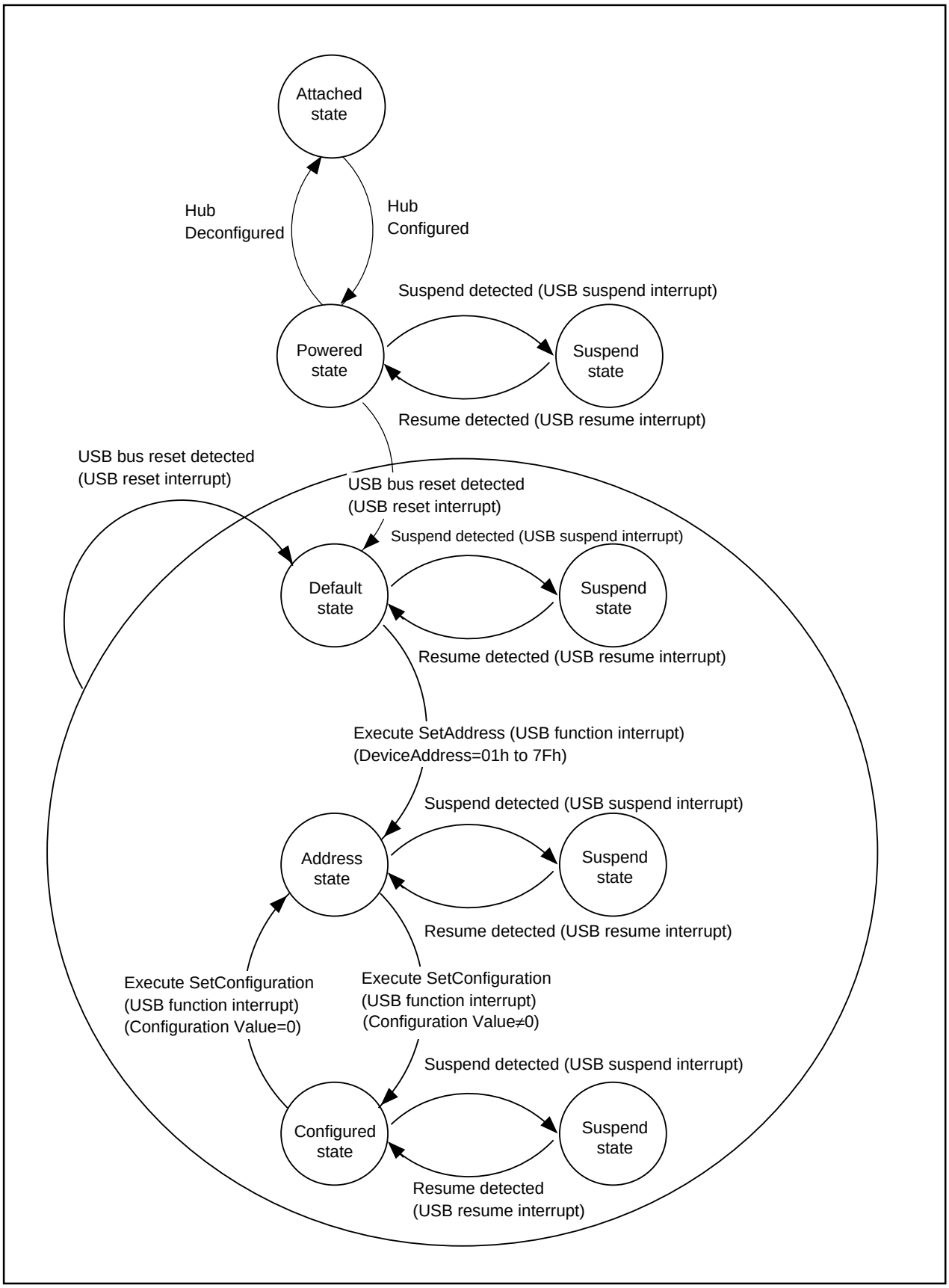

Attached state
Hub Deconfigured
Hub Configured
Suspend detected (USB suspend interrupt)
Powered state
Suspend state
Resume detected (USB resume interrupt)
USB bus reset detected (USB reset interrupt)
USB bus reset detected (USB reset interrupt)
Suspend detected (USB suspend interrupt)
Default state
Suspend state
Resume detected (USB resume interrupt)
Execute SetAddress (USB function interrupt) (DeviceAddress=01h to 7Fh)
Suspend detected (USB suspend interrupt)
Address state
Suspend state
Resume detected (USB resume interrupt)
Execute SetConfiguration (USB function interrupt) (Configuration Value=0)
Execute SetConfiguration (USB function interrupt) (Configuration Value≠0)
Suspend detected (USB suspend interrupt)
Configured state
Suspend state
Resume detected (USB resume interrupt)

Figure 2.8.7. Device state transition

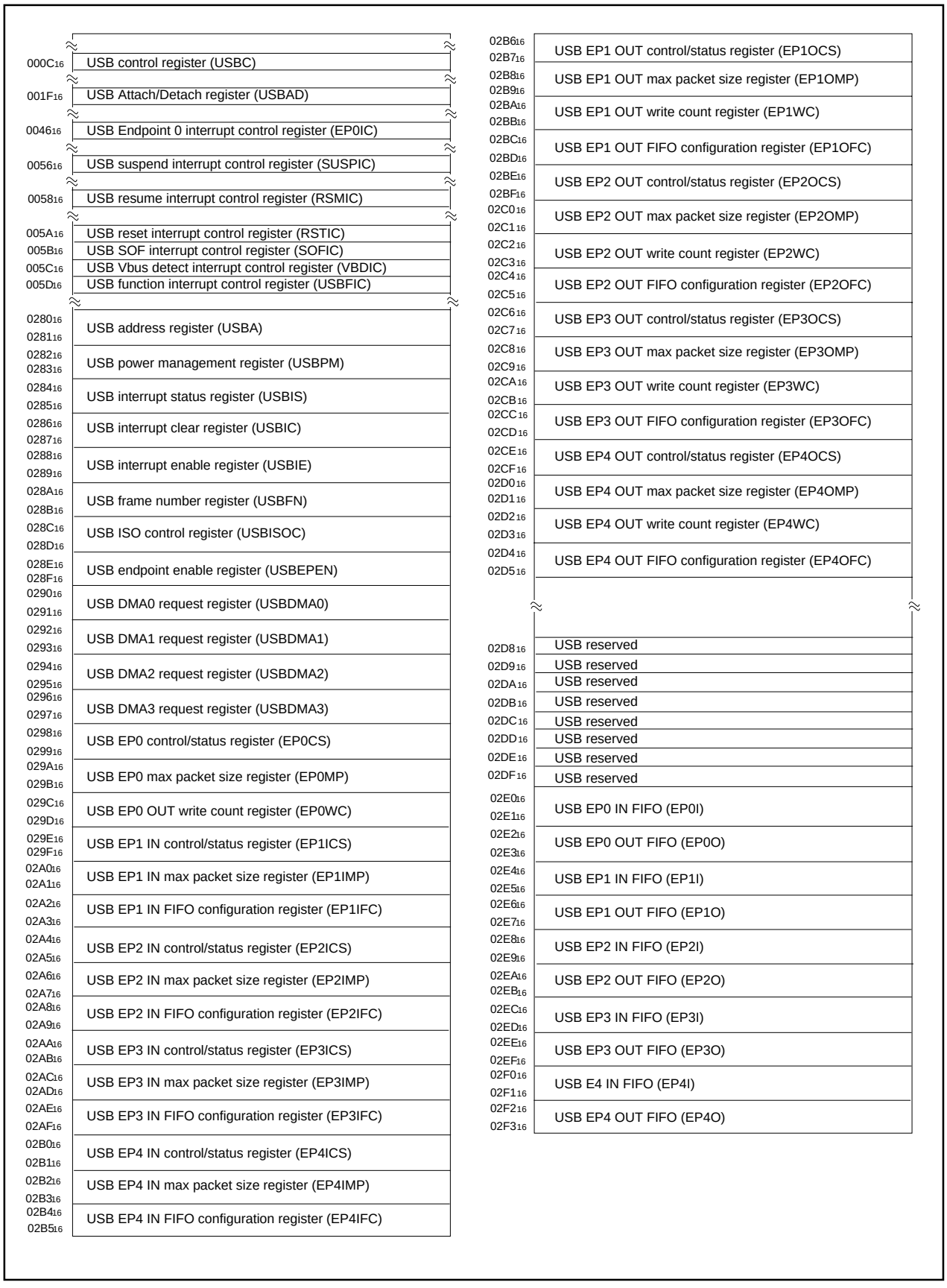

000C16 USB control register (USBC)
001F16 USB Attach/Detach register (USBAD)
004616 USB Endpoint 0 interrupt control register (EP0IC)
005616 USB suspend interrupt control register (SUSPIC)
005816 USB resume interrupt control register (RSMIC)
005A16 USB reset interrupt control register (RSTIC)
005B16 USB SOF interrupt control register (SOFIC)
005C16 USB Vbus detect interrupt control register (VBDIC)
005D16 USB function interrupt control register (USBFIC)
028016 USB address register (USBA)
028116
028216 USB power management register (USBPM)
028316
028416 USB interrupt status register (USBIS)
028516
028616 USB interrupt clear register (USBIC)
028716
028816 USB interrupt enable register (USBIE)
028916
028A16 USB frame number register (USBFN)
028B16
028C16 USB ISO control register (USBISOC)
028D16
028E16 USB endpoint enable register (USBEPEN)
028F16
029016 USB DMA0 request register (USBDMA0)
029116
029216 USB DMA1 request register (USBDMA1)
029316
029416 USB DMA2 request register (USBDMA2)
029516
029616 USB DMA3 request register (USBDMA3)
029716
029816 USB EP0 control/status register (EP0CS)
029916
029A16 USB EP0 max packet size register (EP0MP)
029B16
029C16 USB EP0 OUT write count register (EP0WC)
029D16
029E16 USB EP1 IN control/status register (EP1ICS)
029F16
02A016 USB EP1 IN max packet size register (EP1IMP)
02A116
02A216 USB EP1 IN FIFO configuration register (EP1IFC)
02A316
02A416 USB EP2 IN control/status register (EP2ICS)
02A516
02A616 USB EP2 IN max packet size register (EP2IMP)
02A716
02A816 USB EP2 IN FIFO configuration register (EP2IFC)
02A916
02AA16 USB EP3 IN control/status register (EP3ICS)
02AB16
02AC16 USB EP3 IN max packet size register (EP3IMP)
02AD16
02AE16 USB EP3 IN FIFO configuration register (EP3IFC)
02AF16
02B016 USB EP4 IN control/status register (EP4ICS)
02B116
02B216 USB EP4 IN max packet size register (EP4IMP)
02B316
02B416 USB EP4 IN FIFO configuration register (EP4IFC)
02B516
02B616 USB EP1 OUT control/status register (EP1OCS)
02B716
02B816 USB EP1 OUT max packet size register (EP1OMP)
02B916
02BA16 USB EP1 OUT write count register (EP1WC)
02BB16
02BC16 USB EP1 OUT FIFO configuration register (EP1OFC)
02BD16
02BE16 USB EP2 OUT control/status register (EP2OCS)
02BF16
02C016 USB EP2 OUT max packet size register (EP2OMP)
02C116
02C216 USB EP2 OUT write count register (EP2WC)
02C316
02C416 USB EP2 OUT FIFO configuration register (EP2OFC)
02C516
02C616 USB EP3 OUT control/status register (EP3OCS)
02C716
02C816 USB EP3 OUT max packet size register (EP3OMP)
02C916
02CA16 USB EP3 OUT write count register (EP3WC)
02CB16
02CC16 USB EP3 OUT FIFO configuration register (EP3OFC)
02CD16
02CE16 USB EP4 OUT control/status register (EP4OCS)
02CF16
02D016 USB EP4 OUT max packet size register (EP4OMP)
02D116
02D216 USB EP4 OUT write count register (EP4WC)
02D316
02D416 USB EP4 OUT FIFO configuration register (EP4OFC)
02D516
02D816 USB reserved
02D916 USB reserved
02DA16 USB reserved
02DB16 USB reserved
02DC16 USB reserved
02DD16 USB reserved
02DE16 USB reserved
02DF16 USB reserved
02E016 USB EP0 IN FIFO (EP0I)
02E116
02E216 USB EP0 OUT FIFO (EP0O)
02E316
02E416 USB EP1 IN FIFO (EP1I)
02E516
02E616 USB EP1 OUT FIFO (EP1O)
02E716
02E816 USB EP2 IN FIFO (EP2I)
02E916
02EA16 USB EP2 OUT FIFO (EP2O)
02EB16
02EC16 USB EP3 IN FIFO (EP3I)
02ED16
02EE16 USB EP3 OUT FIFO (EP3O)
02EF16
02F016 USB E4 IN FIFO (EP4I)
02F116
02F216 USB EP4 OUT FIFO (EP4O)
02F316

Figure 2.8.8. USB registers memory mapping

Table 2.8.2. List of USB Related Registers Items

Section name	Register name
2.8.2 USB function control	USB control register, USB Attach/Detach register, USB endpoint enable register, USB endpoint x(x=0-4) IN FIFO data register, USB endpoint x(x=0-4) OUT FIFO data register
2.8.3 USB Interrupt	USB function interrupt status register, USB function interrupt clear register, USB function interrupt enable register, USB frame number register
2.8.4 USB Operation (Suspend/Resume Function)	USB power management register
2.8.5 USB Operation (Endpoint 0)	USB address register, USB endpoint 0 control and status register, USB endpoint 0 MAXP register, USB endpoint 0 OUT write count register
2.8.6 USB Operation (Endpoint 1-4 reception)	USB endpoint x(x=1-4) OUT control and status register, USB endpoint x(x=1-4) OUT MAXP register, USB endpoint x(x=1-4) OUT write count register, USB endpoint x(x=1-4) OUT FIFO configuration register
2.8.7 USB Operation (Endpoint 1-4 transmission)	USB ISO control register, USB endpoint x(x=1-4) IN control and status register, USB endpoint x(x=1-4) IN MAXP register, USB endpoint x(x=1-4) IN FIFO configuration register
2.8.8 USB Operation (Interface of USB and DMAC transfer)	USB DMAx(x=0-3) request register

2.8.2 USB function control

The USB function control unit needs to be enabled for using the USB function. The initialization procedure of the USB function control unit is explained below:

(1) Related Registers

● **USB control register**

This register is used to control each operation of the USB function control unit. When using the USB function, be sure to set USB clock enable bit to "1" before USB enable bit is set to "1". This register is not affected by the USB reset signal. After the USB is enabled (USBC7 ="1"), a minimum 187.5 ns of delay (three cycles of BCLK) is required before performing any other USB register read/write operations.

•USB clock enable bit

This bit is used to enable/disable the USB clock (fusb). This clock is supplied from frequency synthesizer and is required for the USB operation. Set this bit to "1" when enabling the USB clock.

•USB $\overline{\text{SOF}}$ port select bit

This bit is used to enable/disable a SOF signal output on the $P9_2$ pin. Set this bit to "1" when using the USB SOF signal. In this case, set the port $P9_2$ to output mode. When this bit is set to "1", low pulse is always output for about 166ns (2 cycles of the 12MHz USB clock) at start of the frame packet.

•USB enable bit

This bit is used to enable/disable the USB block. When this bit is set to "1", USB function is enabled. After setting "1" to this bit, wait for at least 250ns and, then, read or write the other USB related register.

The configuration of USB control register is shown in Figure 2.8.9.

USB Control register

b7	b6	b5	b4	b3	b2	b1	b0
			0	0	0	0	0

Symbol: USBC | Address: $000C_{16}$ | When reset: 00_{16}

Bit Symbol	Bit Name	Function	R	W
Reserved		Must always be set to "0"	O	O
USBC5	USB clock enable bit	0 : Disable 1 : Enable	O	O
USBC6	USB SOF port select bit	0 : Disable (Note 1) 1 : Enable	O	O
USBC7	USB enable bit	0 : Disable (Note 2) 1 : Enable	O	O

Note 1: $P9_2$ is used as GPI/O pin.
Note 2: All USB internal registers are held at their default values.

Figure 2.8.9. USB control register

● **USB attach/detach register**

This register is used to control attach/detach from the USB host without physically attaching/detaching the USB cable.

•Port 9_0-Second bit

The port $P9_0$ operates as standard port when this bit is set to “0”. Connect a 1.5kΩ resistance between the USB D+ pin and the Uvcc pin. (The timing of D+ line pull-up is depending on the Uvcc pin.)

When this bit is set to “1”, the $P9_0$ has the USB attach/detach function, serving as the power supply pin for pull-up to the D+ line. When using the USB attach/detach function, be sure to connect between the USB D+ pin and the $P9_0$/ATTACH pin via a 1.5kΩ pull-up resistance. Also, in either case, be sure to connect the UVcc pin to the power source.

•Attach/detach bit

This bit is valid when port 9_0-Second bit is set to “1”. When this bit is set to “0”, supply of UVcc pin voltage to $P9_0$ is stopped and the USB cable becomes a detach state artificially. When this bit is set to “1”, the USB cable becomes a attach state artificially since the Uvcc pin voltage is supplied to the $P9_0$ and D+ line is pulled up. After frequency synthesizer is stabilized, set “1” (attach state) to this bit.

•Vbus detect enable bit

This bit is used to enable Vbus detection by setting to “1”. When enabling the Vbus detection, connect the Vbus pin of USB connector to a VbusDTCT pin.

When attach/detach bit is changed from “0” to “1” or from “1” to “0”, time required till the host recognizes attach/detach varies according to board resistance factor/ capacity factor/ USB cable capacity of the device, host's board characteristics, and processing speed. Perform sufficient evaluation through controlling attach/detach bit in accordance with the actual user's system.

The configuration of USB attach/detach register is shown in Figure 2.8.10.

USB Attach/Detach Register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	0	0		

Symbol: USBAD — Address: $001F_{16}$ — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
$P9_0$-second	Port 9_0-Second	0 : Normal mode for Port 9_0 1 : Forces Port 9_0 to operate as pull up for D+.	○	○
Attach/ Detach	Attach/Detach	0 : ATTACH 1 : DETACH	○	○
Reserved		Must always be set to “0”	○	○
VBDT	Vbus detect enable	0 : Disabled 1 : Enabled	○	○

Figure 2.8.10. USB Attach/Detach register

● **USB endpoint enable register**

Endpoints 1 to 4 are used to enable endpoint IN/OUT FIFOs for use. The endpoint 0 is always enabled and cannot be disabled by software. All Endpoints 1 to 4 are disabled after reset.

The configuration of USB endpoint enable register is shown in Figure 2.8.11.

USB Endpoint Enable register

(b15) b7 (b8) b0 b7 b0

0	0	0	0	0	0	0	0							

Symbol: USBEPEN
Address: 028E16
When reset: 000016

Bit Symbol	Bit Name	Function	R	W
EP1_OUT	EP1 OUT enable	0 : Disabled 1 : Enabled	O	O
EP1_IN	EP1 IN enable			
EP2_OUT	EP2 OUT enable			
EP2_IN	EP2 IN enable			
EP3_OUT	EP3 OUT enable			
EP3_IN	EP3 IN enable			
EP4_OUT	EP4 OUT enable			
EP4_IN	EP4 IN enable			
Reserved		Must always be “0”	O	O

Figure 2.8.11. USB endpoint enable register

● **USB endpoint x(x=0 to 4) IN FIFO data register**

Endpoints 0 to 4 respectively have their IN FIFOs. At the time of transmission to the host PC, write the transmit data in these registers. Access these registers in word cycle or byte cycle to the lower byte.

The configuration of USB x(x=0~4) IN FIFO data register is shown in Figure 2.8.12.

USB Endpoint x IN FIFO Data register

(b15) b7 (b8) b0 b7 b0

Symbol: EPxI (x = 0 - 4)
Address: 02E016, 02E416, 02E816, 02EC16, 02F016
When reset: N/A

Bit Symbol	Bit Name	Function	R	W
DATA_15-0	EP0 IN FIFO Data	Write transmit data to this register	X	O

Note 1: Data is undefined if this register is read.
Note 2: Write only to this register with a Word command or a Byte command to the lower 8 bits. Do not write a byte of data to the upper 8 bits. (b8 - b15)

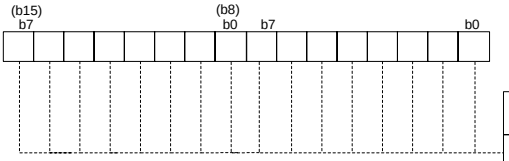
Figure 2.8.12. USB x(x=0~4) IN FIFO data register

● USB endpoint x(x=0 to 4) OUT FIFO data register

Endpoints 0 to 4 respectively have their OUT FIFOs. When data are received from the host PC, read the receive data from these registers. Access these registers in word cycle or byte cycle to the lower byte.

The configuration of USB x(x=0~4) OUT FIFO data register is shown in Figure 2.8.13.

USB Endpoint x OUT FIFO Data register



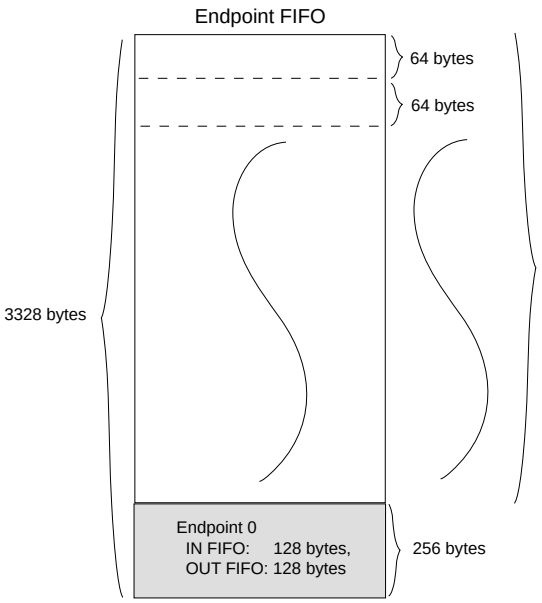
Symbol	Address	When reset
EPxO (x = 0 - 4)	02E2$_{16}$, 02E6$_{16}$, 02EA$_{16}$, 02EE$_{16}$, 02F2$_{16}$	N/A

Bit Symbol	Bit Name	Function	R	W
DATA_15-0	EP0 OUT FIFO Data	Read receive data from this register	O	X

Note 1: Writing to this register might cause a system error.
Note 2: Read only from this register with a Word command or a Byte command to the lower 8 bits. Do not read a byte of data from the upper 8 bits. (b8 - b15)

Figure 2.8.13. USB x(x=0~4) OUT FIFO data register

The endpoint x IN/OUT FIFO mapping is shown in Figure 2.8.14.



This area is allocated for IN/OUT FIFOs of the endpoint 1 to 4.
The FIFO size and start position can be specified for every 64-byte by
USB EPx IN FIFO configuration register and USB EPx OUT FIFO configuration register.

Figure 2.8.14. Endpoint x IN/OUT FIFO mapping

(2) Enable of USB Function Control Unit

The initialization procedure of the USB function control unit of the M30245 group after hardware reset is explained below. Further, for power supply being supplied from the USB, the total driving current has to be controlled to keep equal to or below 100mA.

● Setting of the frequency synthesizer

1: Clear protect.

2: Set the frequency synthesizer related registers to generate 48MHz required for the fUSB.

3: Enable the frequency synthesizer by setting bit 0 in the frequency synthesizer control register to "1".

4: Disable protect.

5: Wait for 3ms to stabilize the frequency synthesizer.

6: Check frequency synthesizer lock status bit. If the frequency synthesizer is in unlock state, waiting for 0.1ms and rechecking are repeated.

● Setting of the USB function control unit

7: Set the USBC5 to "1" to enable the USB clock.

8: Set the USB attach/detach register to "03_{16}" (Port $P9_0$ is set as the power supply pin for pull-up to the D+ line), and set the USBC7 to "1" to enable the USB block. For operation of the USB related registers, a minimum 187.5ns of wait (three cycles of BCLK) is required.

Initialize the endpoint to be used after the USB function control unit being enabled.

The setting example of frequency synthesizer division is shown in Figure 2.8.15. The setup timing of frequency synthesizer after hardware reset is shown in Figure 2.8.16. The initialization procedure of frequency synthesizer and USB function control unit are shown in Figure 2.8.17 and Figure 2.8.18. The initialization procedure of endpoint is shown in Figure 2.8.19 and Figure 2.8.20.

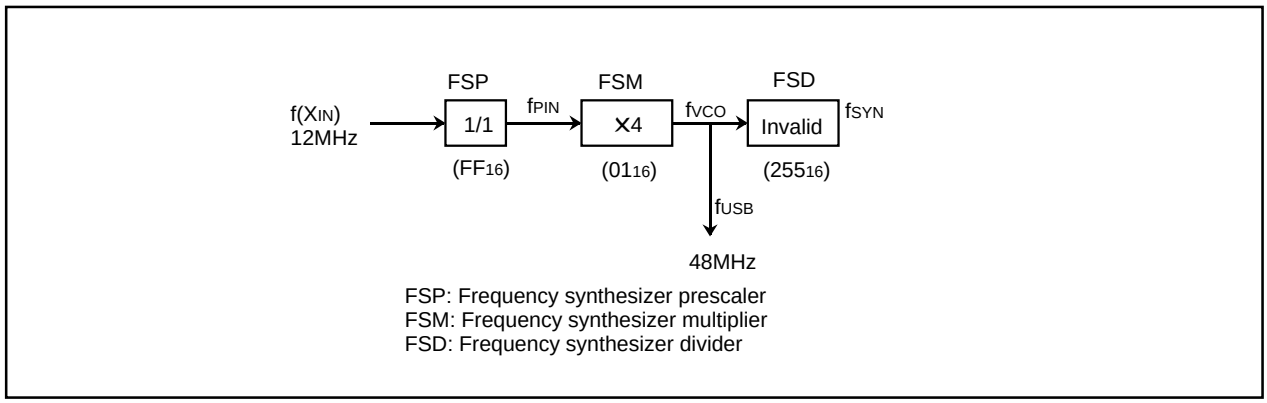


Figure 2.8.15. Setting example of frequency synthesizer division

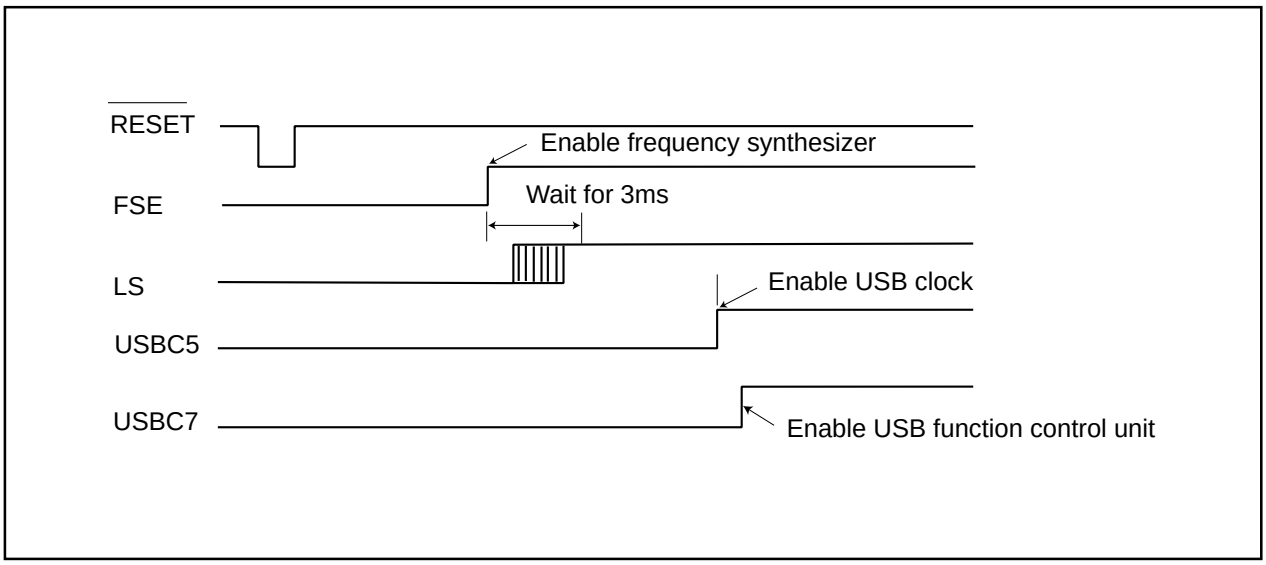


Figure 2.8.16. Setup timing of frequency synthesizer after hardware reset

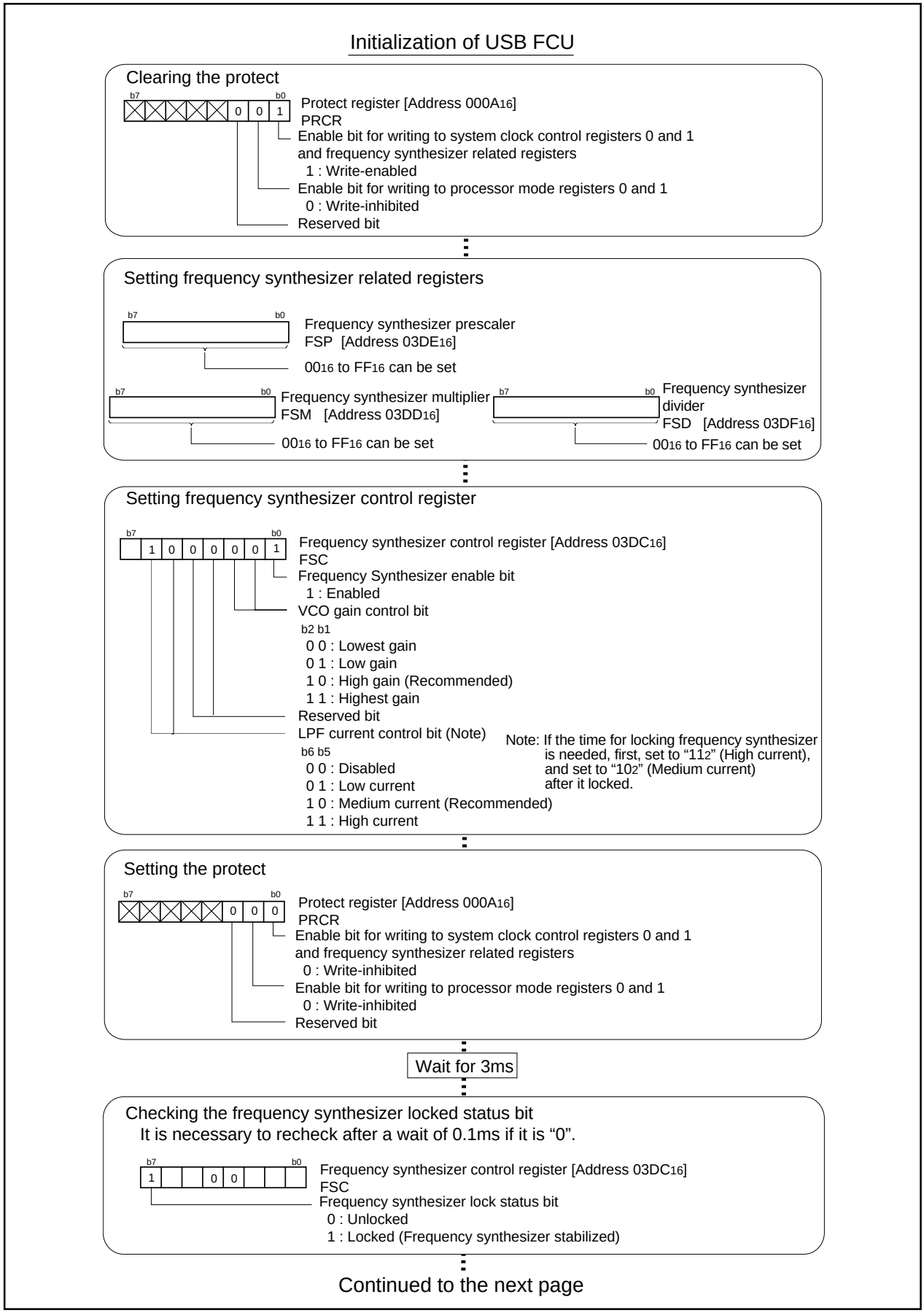

Initialization of USB FCU
Clearing the protect
b7 b0
0 0 1
Protect register [Address 000A16]
PRCR
Enable bit for writing to system clock control registers 0 and 1
and frequency synthesizer related registers
1 : Write-enabled
Enable bit for writing to processor mode registers 0 and 1
0 : Write-inhibited
Reserved bit
Setting frequency synthesizer related registers
Frequency synthesizer prescaler
FSP [Address 03DE16]
0016 to FF16 can be set
Frequency synthesizer multiplier
FSM [Address 03DD16]
0016 to FF16 can be set
Frequency synthesizer
divider
FSD [Address 03DF16]
0016 to FF16 can be set
Setting frequency synthesizer control register
1 0 0 0 0 0 1
Frequency synthesizer control register [Address 03DC16]
FSC
Frequency Synthesizer enable bit
1 : Enabled
VCO gain control bit
b2 b1
0 0 : Lowest gain
0 1 : Low gain
1 0 : High gain (Recommended)
1 1 : Highest gain
Reserved bit
LPF current control bit (Note)
b6 b5
0 0 : Disabled
0 1 : Low current
1 0 : Medium current (Recommended)
1 1 : High current
Note: If the time for locking frequency synthesizer is needed, first, set to "112" (High current), and set to "102" (Medium current) after it locked.
Setting the protect
0 0 0
Protect register [Address 000A16]
PRCR
Enable bit for writing to system clock control registers 0 and 1
and frequency synthesizer related registers
0 : Write-inhibited
Enable bit for writing to processor mode registers 0 and 1
0 : Write-inhibited
Reserved bit
Wait for 3ms
Checking the frequency synthesizer locked status bit
It is necessary to recheck after a wait of 0.1ms if it is "0".
1 0 0
Frequency synthesizer control register [Address 03DC16]
FSC
Frequency synthesizer lock status bit
0 : Unlocked
1 : Locked (Frequency synthesizer stabilized)
Continued to the next page

Figure 2.8.17. Initialization procedure of frequency synthesizer and USB function control unit (1)

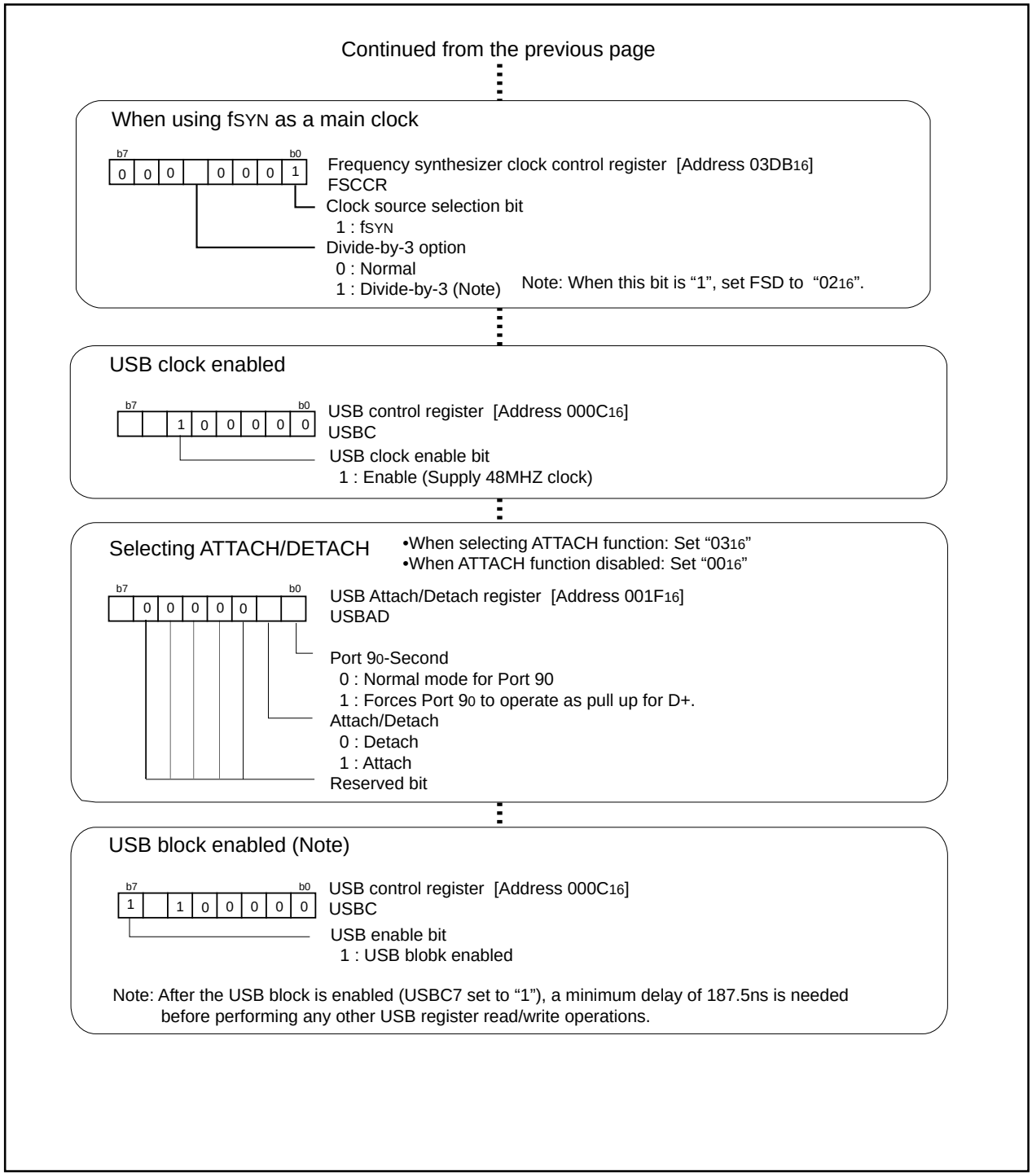

Continued from the previous page
When using fSYN as a main clock
b7 b0
0 0 0 0 0 0 1
Frequency synthesizer clock control register [Address 03DB16]
FSCCR
Clock source selection bit
1 : fSYN
Divide-by-3 option
0 : Normal
1 : Divide-by-3 (Note)
Note: When this bit is "1", set FSD to "0216".
USB clock enabled
b7 b0
1 0 0 0 0 0
USB control register [Address 000C16]
USBC
USB clock enable bit
1 : Enable (Supply 48MHZ clock)
Selecting ATTACH/DETACH
•When selecting ATTACH function: Set "0316"
•When ATTACH function disabled: Set "0016"
b7 b0
0 0 0 0 0
USB Attach/Detach register [Address 001F16]
USBAD
Port 90-Second
0 : Normal mode for Port 90
1 : Forces Port 90 to operate as pull up for D+.
Attach/Detach
0 : Detach
1 : Attach
Reserved bit
USB block enabled (Note)
b7 b0
1 1 0 0 0 0 0
USB control register [Address 000C16]
USBC
USB enable bit
1 : USB blobk enabled
Note: After the USB block is enabled (USBC7 set to "1"), a minimum delay of 187.5ns is needed
before performing any other USB register read/write operations.

Figure 2.8.18. Initialization procedure of frequency synthesizer and USB function control unit (2)

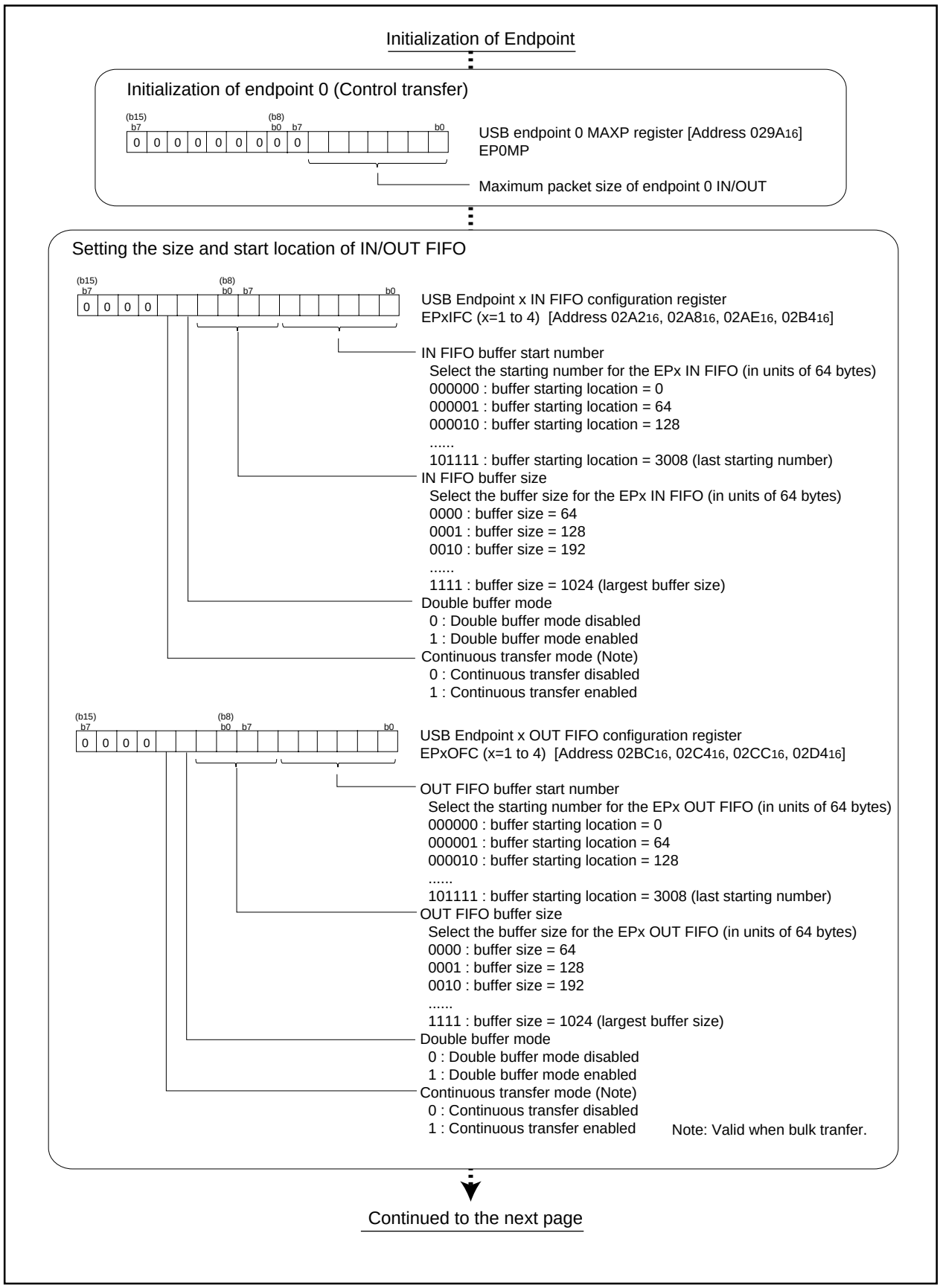

Initialization of Endpoint
Initialization of endpoint 0 (Control transfer)
(b15) b7 (b8) b0 b7 b0
0 0 0 0 0 0 0 0 0
USB endpoint 0 MAXP register [Address 029A16]
EP0MP
Maximum packet size of endpoint 0 IN/OUT
Setting the size and start location of IN/OUT FIFO
(b15) b7 (b8) b0 b7 b0
0 0 0 0
USB Endpoint x IN FIFO configuration register
EPxIFC (x=1 to 4) [Address 02A216, 02A816, 02AE16, 02B416]
IN FIFO buffer start number
Select the starting number for the EPx IN FIFO (in units of 64 bytes)
000000 : buffer starting location = 0
000001 : buffer starting location = 64
000010 : buffer starting location = 128
.....
101111 : buffer starting location = 3008 (last starting number)
IN FIFO buffer size
Select the buffer size for the EPx IN FIFO (in units of 64 bytes)
0000 : buffer size = 64
0001 : buffer size = 128
0010 : buffer size = 192
.....
1111 : buffer size = 1024 (largest buffer size)
Double buffer mode
0 : Double buffer mode disabled
1 : Double buffer mode enabled
Continuous transfer mode (Note)
0 : Continuous transfer disabled
1 : Continuous transfer enabled
(b15) b7 (b8) b0 b7 b0
0 0 0 0
USB Endpoint x OUT FIFO configuration register
EPxOFC (x=1 to 4) [Address 02BC16, 02C416, 02CC16, 02D416]
OUT FIFO buffer start number
Select the starting number for the EPx OUT FIFO (in units of 64 bytes)
000000 : buffer starting location = 0
000001 : buffer starting location = 64
000010 : buffer starting location = 128
.....
101111 : buffer starting location = 3008 (last starting number)
OUT FIFO buffer size
Select the buffer size for the EPx OUT FIFO (in units of 64 bytes)
0000 : buffer size = 64
0001 : buffer size = 128
0010 : buffer size = 192
.....
1111 : buffer size = 1024 (largest buffer size)
Double buffer mode
0 : Double buffer mode disabled
1 : Double buffer mode enabled
Continuous transfer mode (Note)
0 : Continuous transfer disabled
1 : Continuous transfer enabled
Note: Valid when bulk tranfer.
Continued to the next page

Figure 2.8.19. Initialization procedure of endpoint (1)

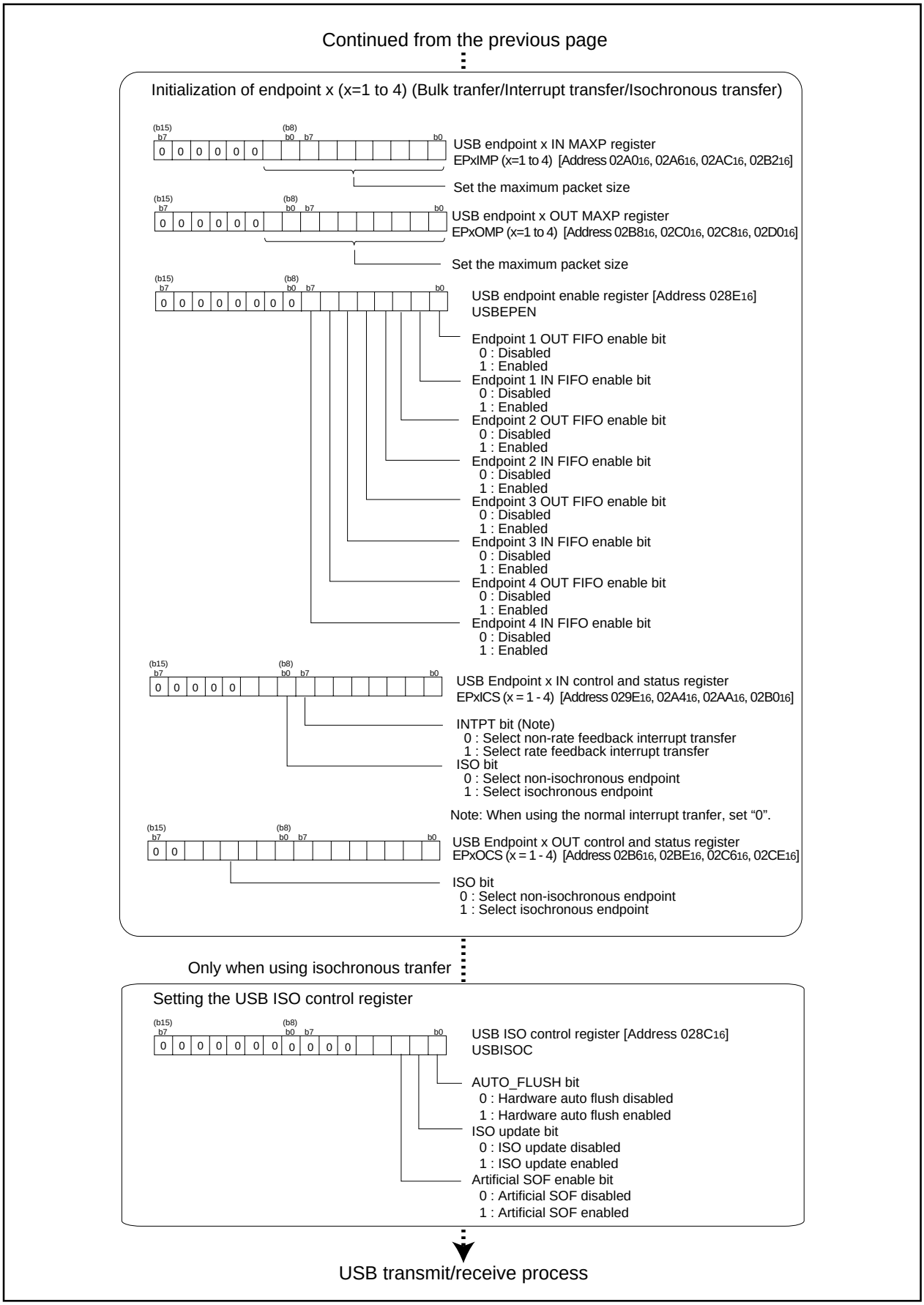

Continued from the previous page
Initialization of endpoint x (x=1 to 4) (Bulk tranfer/Interrupt transfer/Isochronous transfer)
USB endpoint x IN MAXP register
EPxIMP (x=1 to 4) [Address 02A016, 02A616, 02AC16, 02B216]
Set the maximum packet size
USB endpoint x OUT MAXP register
EPxOMP (x=1 to 4) [Address 02B816, 02C016, 02C816, 02D016]
Set the maximum packet size
USB endpoint enable register [Address 028E16]
USBEPEN
Endpoint 1 OUT FIFO enable bit
0 : Disabled
1 : Enabled
Endpoint 1 IN FIFO enable bit
0 : Disabled
1 : Enabled
Endpoint 2 OUT FIFO enable bit
0 : Disabled
1 : Enabled
Endpoint 2 IN FIFO enable bit
0 : Disabled
1 : Enabled
Endpoint 3 OUT FIFO enable bit
0 : Disabled
1 : Enabled
Endpoint 3 IN FIFO enable bit
0 : Disabled
1 : Enabled
Endpoint 4 OUT FIFO enable bit
0 : Disabled
1 : Enabled
Endpoint 4 IN FIFO enable bit
0 : Disabled
1 : Enabled
USB Endpoint x IN control and status register
EPxICS (x = 1 - 4) [Address 029E16, 02A416, 02AA16, 02B016]
INTPT bit (Note)
0 : Select non-rate feedback interrupt transfer
1 : Select rate feedback interrupt transfer
ISO bit
0 : Select non-isochronous endpoint
1 : Select isochronous endpoint
Note: When using the normal interrupt tranfer, set “0”.
USB Endpoint x OUT control and status register
EPxOCS (x = 1 - 4) [Address 02B616, 02BE16, 02C616, 02CE16]
ISO bit
0 : Select non-isochronous endpoint
1 : Select isochronous endpoint
Only when using isochronous tranfer
Setting the USB ISO control register
USB ISO control register [Address 028C16]
USBISOC
AUTO_FLUSH bit
0 : Hardware auto flush disabled
1 : Hardware auto flush enabled
ISO update bit
0 : ISO update disabled
1 : ISO update enabled
Artificial SOF enable bit
0 : Artificial SOF disabled
1 : Artificial SOF enabled
USB transmit/receive process

Figure 2.8.20. Initialization procedure of endpoint (1)

(3) Disable of USB Function Control Unit

After the USB function control unit being enabled, if the system design requires to disable the USB function, follow the procedure below:

1: Disable the USB clock by clearing USB enable bit (USBC7) to “0”.
2: Disable the USB clock by clearing USB clock enable bit (USBC5) to “0”.
3: Disable the frequency synthesizer by clearing frequency synthesizer enable bit (FSE) to “0”.

Normally, for system design which continues enabling the USB function, disabling the USB function control unit is not required.

(4) Vbus Detection

During USB self-powered operation, the Vbus detect function is used to switch into bus power only when the device is connected to the host PC and power supply is available from the Vbus, for minimizing battery consumption. To use the Vbus detect function, it is necessary that the VbusDTCT pin is processed by hardware and the Vbus detect interrupt is set by software. The VbusDTCT pin is used for the Vbus detect function. When operating the USB in self-powered mode, connect the Vbus line from the USB connector to the VbusDTCT pin. For enable/disable of the Vbus detect function, set Vbus detect enable bit (bit 7 at address $001F_{16}$) of USB attach/detach register to “1”. Set the interrupt priority level by using USB Vbus detect interrupt control register (VBDIC: address $005C_{16}$). Each time the USB host powers ON/OFF, a Vbus detect interrupt will be occurred. When a Vbus detect interrupt is occurred, the Vbus detect state bit located in the port 9 data register (bit 1 at address $03F1_{16}$) should be read to determine if the Vbus is powered ON/OFF.

To avoid receiving a false Vbus detect interrupt at start-up, the Vbus detect should be enabled before enabling the Vbus detect interrupt. Use the following procedure when enabling the Vbus detect function:

1: Enable a Vbus detect by setting “1” to Vbus detect enable bit (bit 7 at address $001F_{16}$).
2: Clear the Vbus detect interrupt request by setting “0” to Vbus detect interrupt request bit (bit 3 at address $005C_{16}$).
3: Enable the Vbus detect interrupt by setting the Vbus detect interrupt priority level greater than “000_2” (bit 0 to 2 at address $005C_{16}$).

2.8.3 USB Interrupt

The USB related interrupts include USB suspend interrupt, USB resume interrupt, USB reset interrupt, USB endpoint 0 interrupt, USB function interrupt, and USB SOF interrupt.

(1) Related Registers

● **USB function interrupt status register**

This register is used to judge the USB function interrupt factor. This is the read-only register which indicates each interrupt request state of endpoint x(x=1~4) IN interrupt, endpoint x(x=1~4) OUT interrupt, and error interrupt. On occurrence of an interrupt request, this is set to “1”. Each interrupt status flag can be cleared to “0” by setting “1” to the corresponding bit of USB function interrupt clear register.

- **Endpoint 1 IN Interrupt Status Flag**
- **Endpoint 2 IN Interrupt Status Flag**
- **Endpoint 3 IN Interrupt Status Flag**
- **Endpoint 4 IN Interrupt Status Flag**

These flags indicate endpoint x(x=1~4) IN interrupt request state, respectively. These flags are set to “1” at the corresponding endpoint that has been enabled by USB function interrupt enable register in the following cases:

- The endpoint is enabled from a disabled state
- One buffer data is successfully transmitted
- Buffer flush has been executed by either AUTO_FLUSH or FLUSH bit (bit 6 at address EPxICS) being set to “1” while buffer data exists in the IN FIFO.

- **Endpoint 1 OUT Interrupt Status Flag**
- **Endpoint 2 OUT Interrupt Status Flag**
- **Endpoint 3 OUT Interrupt Status Flag**
- **Endpoint 4 OUT Interrupt Status Flag**

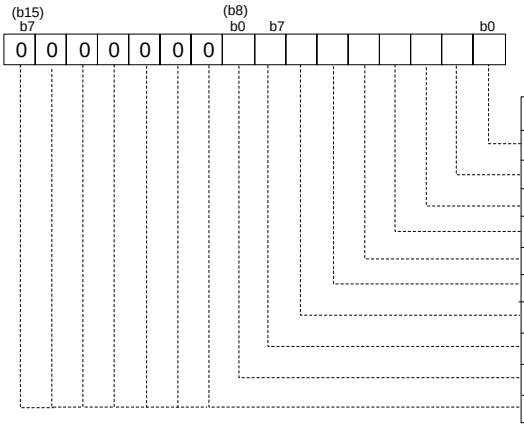
These each indicate endpoint x(x=1~4) OUT interrupt request state. When a data is successfully received, these flags are set to “1” at the corresponding endpoint.

- **Error Interrupt Status Flag**

This flag indicates that an error has been occurred at the endpoint x(x=0~4). This flag is set to “1” in the following cases:

- The EP0CSR4 (FORCE_STALL) flag of endpoint 0 is set to “1”
- The EP0CSR5 (SETUP_END) flag of endpoint 0 is set to “1”
- The INxCSR2 (UNDER_RUN) flag of endpoint 1 to 4 IN is set to “1”
- The OUTxCSR2 (OVER_RUN) flag of endpoint 1 to 4 OUT is set to “1”
- The OUTxCSR3 (FORCE_STALL) flag of endpoint 1 to 4 OUT is set to “1”
- The OUTxCSR4 (DATA_ERR) flag of endpoint 1 to 4 OUT is set to “1”

USB Interrupt Status register (Note)



Symbol	Address	When reset
USBIS	028416	000016

Bit Symbol	Bit Name	Function	R	W
INTST0	EP1 IN interrupt status flag	0 : No interrupt request 1 : Interrupt request issued	O	×
INTST1	EP1 OUT interrupt status flag			
INTST2	EP2 IN interrupt status flag			
INTST3	EP2 OUT interrupt status flag			
INTST4	EP3 IN interrupt status flag			
INTST5	EP3 OUT interrupt status flag			
INTST6	EP4 IN interrupt status flag			
INTST7	EP4 OUT interrupt status flag			
INTST8	Error interrupt status flag			
Reserved		Must always be “0”	O	×

Note: Read only

Figure 2.8.21. USB function interrupt status register

● USB function interrupt clear register

This register is used to clear the USB function interrupt request factor.

The interrupt status flag corresponding to USB function interrupt status register is cleared to “0” by setting “1” to the interrupt status clear flag.

The configuration of USB function interrupt status register is shown in Figure 2.8.22.

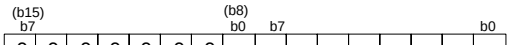
● USB function interrupt enable register

This register is used to set the USB function interrupt request factor.

The USB function interrupt request occurs when the request of interrupt which set the enable bit to “1” occurs.

The configuration of USB function interrupt enable register is shown in Figure 2.8.23.

USB Function Interrupt Clear register



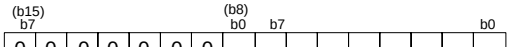
Symbol: USBIC | Address: 028616 | When reset: 000016

Bit Symbol	Bit Name	Function	R	W
INTCL0	Clear EP1 IN interrupt status flag	0 : No action 1 : Clear interrupt status flag	×	O
INTCL1	Clear EP1 OUT interrupt status flag			
INTCL2	Clear EP2 IN interrupt status flag			
INTCL3	Clear EP2 OUT interrupt status flag			
INTCL4	Clear EP3 IN interrupt status flag			
INTCL5	Clear EP3 OUT interrupt status flag			
INTCL6	Clear EP4 IN interrupt status flag			
INTCL7	Clear EP4 OUT interrupt status flag			
INTCL8	Clear error interrupt status flag			
Reserved		Must always be “0”	×	O

Note: Write only

Figure 2.8.22. USB function interrupt clear register

USB Function Interrupt Enable register



Symbol: USBIE | Address: 028816 | When reset: 01FF16

Bit Symbol	Bit Name	Function	R	W
INTEN0	EP1 IN interrupt enable bit	0 : Disabled 1 : Enabled	O	O
INTEN1	EP1 OUT interrupt enable bit			
INTEN2	EP2 IN interrupt enable bit			
INTEN3	EP2 OUT interrupt enable bit			
INTEN4	EP3 IN interrupt enable bit			
INTEN5	EP3 OUT interrupt enable bit			
INTEN6	EP4 IN interrupt enable bit			
INTEN7	EP4 OUT interrupt enable bit			
INTEN8	Error interrupt enable bit			
Reserved		Must always be “0”	O	O

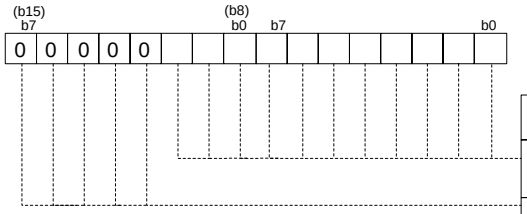
Figure 2.8.23. USB function interrupt enable register

● **USB frame number register**

This register is used to contain 11-bit frame number of SOF token received from the host CPU. This is the read-only register.

The configuration of USB frame number register is shown in Figure 2.8.24.

USB Frame Number register (Note)



Symbol: USBFN | Address: 028A16 | When reset: 000016

Bit Symbol	Bit Name	Function	R	W
FN10-0	SOF frame number bit	11-bit frame number issued with an SOF packet	O	X
Reserved		“0” when read	O	X

Note: Read only.

Figure 2.8.24. USB frame number register

(2) USB Endpoint 0 Interrupt

In the endpoint 0 interrupt, the interrupt request occurs when the data transmit/receive of endpoint 0 are completed. Set the interrupt priority level by using USB endpoint 0 interrupt control register (EP0IC: address 0046_{16}). The interrupt request bit of the EP0IC is set to “1” and the USB endpoint 0 interrupt occurs when one of the following events occur:

- A data is successfully received
- A data is successfully transmitted
- The DATA_END bit of the EP0CS register is cleared to “0”
- The SETUP_END flag of the EP0CS register is set to “1”

● **Mask Function of Endpoint 0 Interrupt Factor**

By setting the DATA_END_MASK bit of USB endpoint 0 control and status register, the M30245 group can control whether or not to clear the DATA_END flag as the endpoint 0 interrupt factor. Clearing of the DATA_END flag is masked at the time of resetting. (The DATA_END flag is not cleared as an endpoint 0 interrupt factor.)

(3) USB Function Interrupt

The USB function interrupts include the endpoint x(x=1~4) IN interrupt, endpoint x(x=1~4) OUT interrupt, and error interrupt. An interrupt request occurs on completion of data transmit/receive or on occurrence of an error such as overrun/underrun, setting the status flag which is the factor of the interrupt request inside USB function interrupt status register to “1”. When using the USB function interrupt, set the interrupt priority level at USB function interrupt control register (address $005D_{16}$) and the corresponding bit of USB function interrupt enable register to “1”.

The USB function interrupt involves multiple interrupt request factors. Therefore, during processing of the USB function interrupt, an interrupt request may occur newly and the interrupt status flag can be changed by it. When performing USB function interrupt processing, be sure to first save contents of interrupt status register and to clear the status flag. Then, process the interrupt request that has occurred when the interrupt process has been received based on the saved data value.

● Endpoint x(x=1~4) IN Interrupt

In the endpoint x(x=1~4) IN interrupt, when each USB endpoint x IN interrupt status flag (INTST0,2,4,6) of the corresponding endpoints of USB function interrupt status register is set to “1”, an interrupt request occurs. Each flag INTST0, 2, 4, 6 is set to “1” in one of the following cases:

- The corresponding bit of USB endpoint enable register (USBEPEN: address $028E_{16}$) is set to “1”. (The endpoint is enabled from a disabled state.)
- A data is successfully transmitted
- AUTO FLUSH of hardware has been executed or FLUSH bit of corresponding USB endpoint x IN control and status register (EPxICS: addresses $029E_{16}$, $02A4_{16}$, $02AA_{16}$, $02B0_{16}$) being set to “1” while one or two packet data exist in the IN FIFO.
- The last ACK for control read transfer is destroyed.

● Endpoint x(x=1~4) OUT Interrupt

In the endpoint x(x=1~4) OUT interrupt, when each USB endpoint x OUT interrupt status flag (INTST1,3,5,7) of the corresponding endpoints of USB function interrupt status register is set to “1”, an interrupt request occurs. When a data is successfully received at the corresponding endpoint, each flag INTST1, 3, 5, 7 is set to “1”.

● Error Interrupt

In the error interrupt, when the error interrupt status flag (INTST8) of USB function interrupt status register is set to “1”, an interrupt request occurs. The INTST8 is set to “1” in one of the following cases:

- The FORCE_STALL flag of endpoint 0 control and status register (EP0CS) is set to “1”.
- The SETUP_END flag of EP0CS is set to “1”.
- The UNDER_RUN flag of USB endpoint x IN control and status register (EPxICS: addresses $029E_{16}$, $02A4_{16}$, $02AA_{16}$, $02B0_{16}$) is set to “1”. (Due to delay in writing of data to FIFO, underrun has occurred at any one of the IN endpoints that are used for isochronous transfer.)
- The OVER_RUN flag of USB endpoint x OUT control and status register (EPxOCS: addresses $02B6_{16}$, $02BE_{16}$, $02C6_{16}$, $02CE_{16}$) is set to “1”. (Due to delay in reading of data from FIFO, overrun has occurred at any one of the OUT endpoints that are used for isochronous transfer.)
- The FORCE_STALL flag of EPxOCS is set to “1”.
- The DATA_ERR flag of EPxOCS is set to “1”.

(4) USB Reset Interrupt

This interrupt is used for detection of the USB reset. This occurs when the USB function control unit has received reset signal from the host CPU (or detected SE0 on the D+/D- line for at least 2.5μs). At this time, all the USB internal registers are made into reset state. To resume communication, each endpoint needs to be initialized. When using the USB reset interrupt, set the interrupt priority level at USB reset interrupt control register (RSTIC: address $005A_{16}$).

(5) USB Suspend Interrupt

This interrupt is used for detection of inactivity on the USB bus line. The suspend status flag of USB power management register (USBPM: address 0282_{16}) is set when the USB function control unit has detected suspend signal on the USB bus line (or not detected any bus activity on the D+/D- line for at least 3ms). Simultaneously, the USB suspend interrupt occurs.

When using the USB suspend interrupt, set the interrupt priority level at USB suspend interrupt control register (SUSPIC: address 0056_{16}).

(6) USB Resume Interrupt

This interrupt is used for detection of activity on the USB bus line while the device is in suspend state. When the USB function control unit has received resume signal on the USB bus line (or detected activity on the D+/D- line), the interrupt request occurs, and the USB resume interrupt occurs by setting “1” to resume interrupt request bit of USB resume interrupt control register (RSMIC: address 0058_{16}).

When using the USB suspend interrupt, set the interrupt priority level at the RSMIC.

(7) USB SOF (Start of Frame) Interrupt

This interrupt is valid to control the isochronous transfer. When a valid SOF PID is detected, receive of an SOF packet is recognized and an interrupt request occurs. The frame number (11 bits) of the SOF packet received from the host is automatically stored in USB frame number register.

In isochronous transfer, $P9_2$ can be used as the SOF output pin by setting “1” to USB SOF port select bit of USB control register. (Set $P9_2$ to output port.) Every time that the SOF signal is received from the host, the $P9_2$ outputs “Low” for about 166ns (two cycles of the 12MHz USB clock.).

When using the USB SOF interrupt, set the interrupt priority level at USB SOF interrupt control register (SOFIC: address $005B_{16}$).

● **Artificial SOF Function**

If the SOF packet from the host PC is destroyed due to any cause, and thus the SOF packet is not correctly received within 1 ms of the start of the frame, a virtual SOF receive operation is performed (and an USB SOF interrupt is generated). Even if the SOF packet is destroyed due to any cause, a new frame can be formed by this function without waiting for the next SOF packet. This function operates once to virtually receive an SOF packet after two SOF packets have been received correctly.

(8) USB Function Interrupt Routine

This interrupt is used to control data flow. This occurs on completion of data transmit/receive or on occurrence of an error such as overrun/underrun. When using the USB function interrupt, set the interrupt priority level at USB function interrupt control register (address 005D16) and the corresponding bit of USB function interrupt enable register to “1”.

The related registers are shown in Figure 2.8.25, and the USB function interrupt routine is shown in Figure 2.8.26.

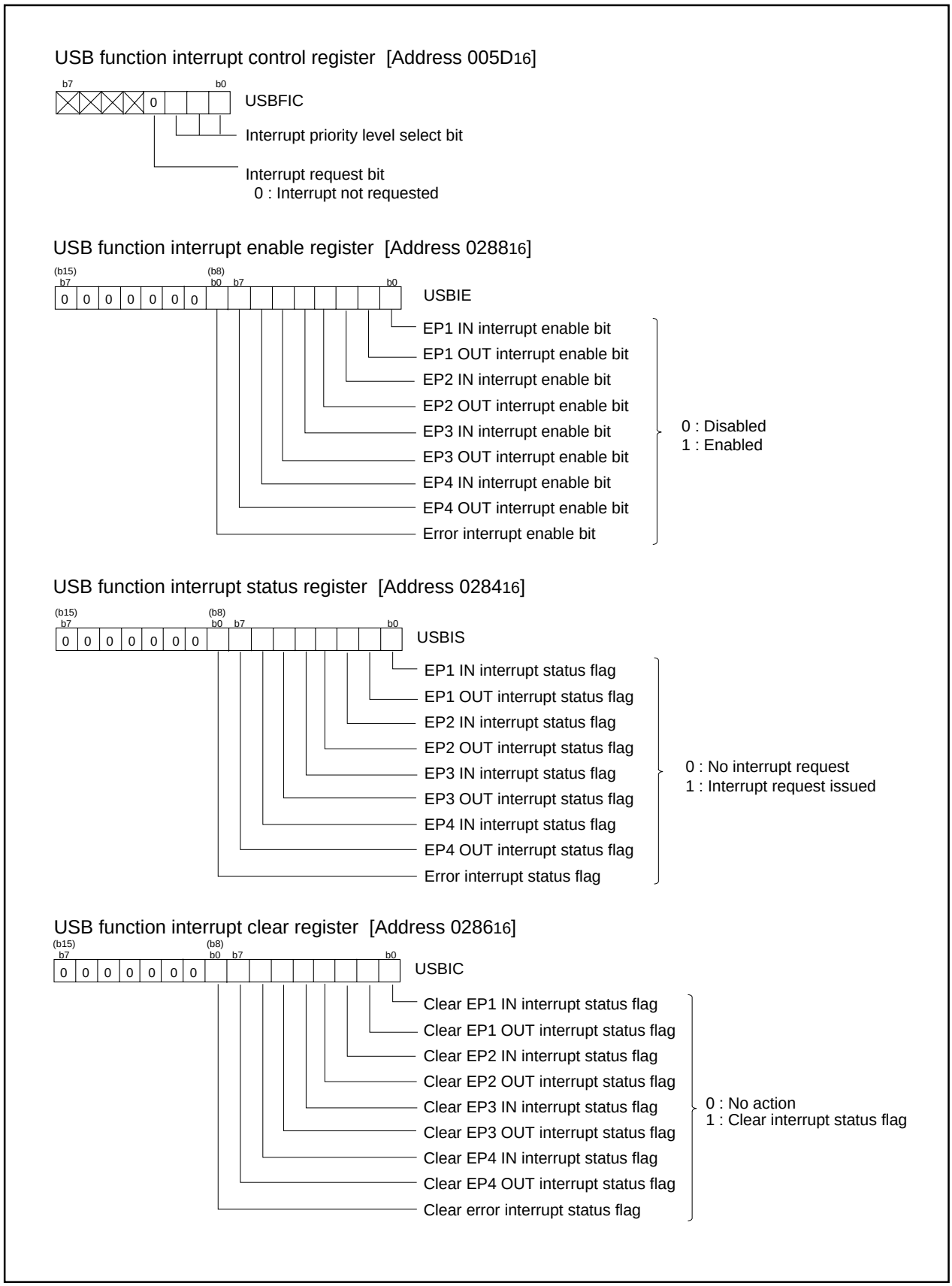


Figure 2.8.25. USB function interrupt related registers

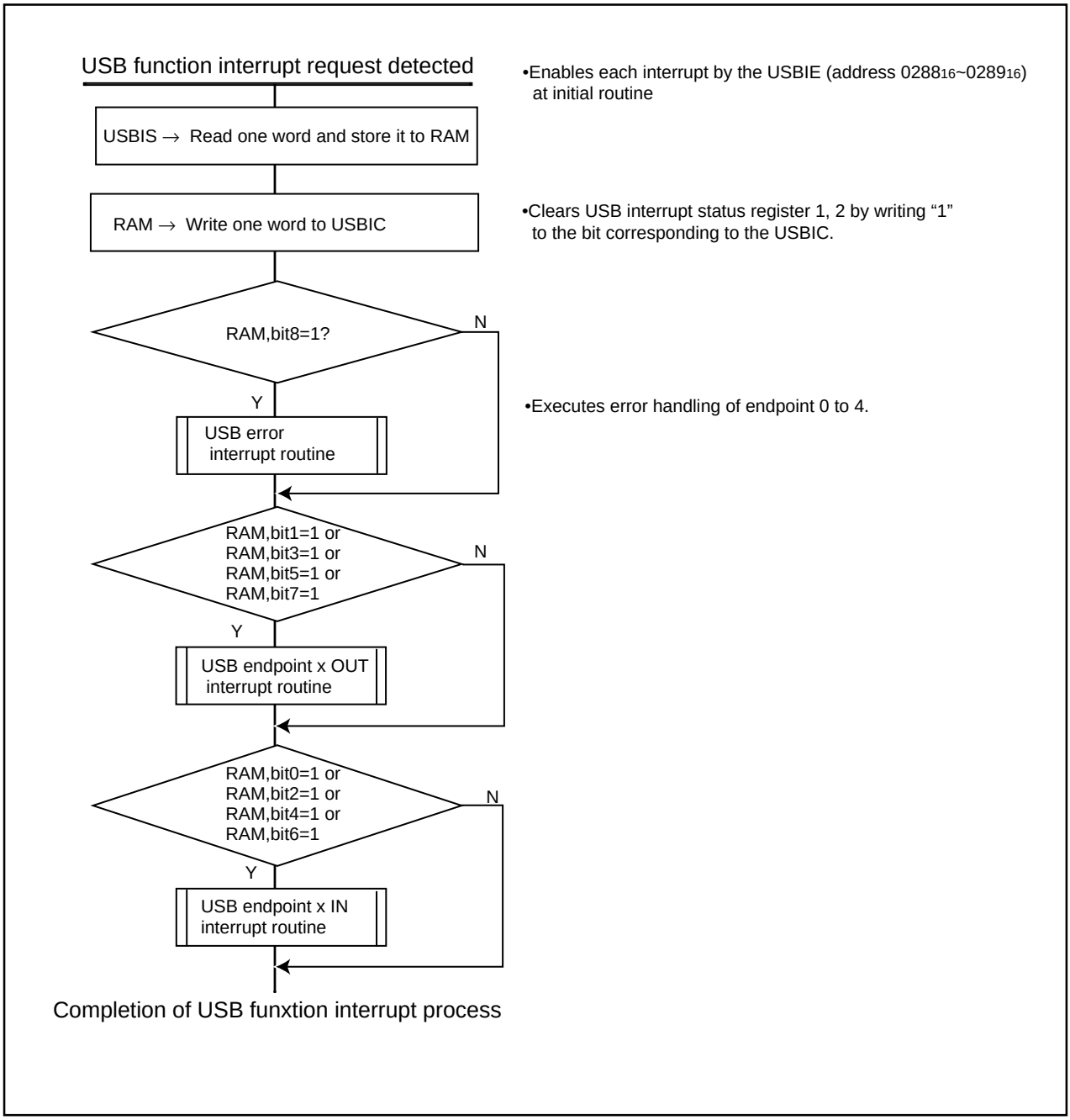

USB function interrupt request detected
USBIS → Read one word and store it to RAM
RAM → Write one word to USBIC
RAM,bit8=1?
N
Y
USB error interrupt routine
RAM,bit1=1 or RAM,bit3=1 or RAM,bit5=1 or RAM,bit7=1
N
Y
USB endpoint x OUT interrupt routine
RAM,bit0=1 or RAM,bit2=1 or RAM,bit4=1 or RAM,bit6=1
N
Y
USB endpoint x IN interrupt routine
Completion of USB funxtion interrupt process
•Enables each interrupt by the USBIE (address 0288₁₆~0289₁₆) at initial routine
•Clears USB interrupt status register 1, 2 by writing “1” to the bit corresponding to the USBIC.
•Executes error handling of endpoint 0 to 4.

Figure 2.8.26. USB function interrupt processing routine

2.8.4 USB Operation (Suspend/Resume Function)

The USB device has received the suspend signal from the host CPU following the power input state, and then controls power supply and shifts the state into the suspend state.

And, by receiving the resume signal from the host CPU (or transmitting the resume signal to the host CPU in the case of remote wakeup), it returns to the state before shifting into the suspend state and resumes the USB communication.

This section explains how the M30245 group controls a shift into suspend state/recovery at the time of resume in the state which the USB function control unit is enabled.

(1) Related Registers

● **USB power management register**

This register is used to control the suspend/resume by the USB function control unit.

- **USB Suspend Status Flag**

When the USB function control unit does not detected any bus activity on D+/D- line for at least 3ms, the USB suspend status flag is set. Simultaneously, the USB suspend interrupt request occurs. This flag is automatically cleared in the following cases:

- The active signal from the host CPU has been detected on the USB's D+/D- line. (When the resume signal has been received and, simultaneously, the USB resume interrupt request has occurred.)
- Transmission of the resume signal to the host CPU has been completed. (After USB remote wakeup bit being set to “1”, when clearing it to “0” to stop resume signal transmission.)

If the USB clock has been disabled during the suspend mode, this flag is not cleared until after the USB clock is re-enabled.

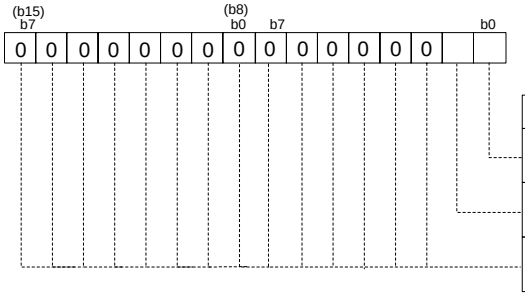
- **USB remote wakeup bit**

When the USB suspend signal status flag is set to “1”, the USB function control unit transmits the resume signal to the host CPU while setting “1” to USB remote wakeup bit.

Set USB remote wakeup bit to “1” in order to transmit the resume signal to the host CPU and to return to the previous state (remote wakeup) in the USB suspend state. Retain this bit at “1” for min. 1ms to max. 15ms before completing the resume signal transmission by clearing to “0”.

The configuration of USB power management register is shown in Figure 2.8.27.

USB Power Management register



Symbol	Address	When reset
USBPM	028216	000016

Bit Symbol	Bit Name	Function	R	W
SUSPEND	Suspend state flag	0 : Not in suspend state 1 : In suspend state (Note 1)	O	× (Note 2)
WAKEUP	Remote wakeup	0 : End remote wakeup signal 1 : Remote wakeup signaling (Note 3)	O	O
Reserved		Must always be “0”	O	O

Note 1: This flag is cleared when WAKEUP=“1” or a resume signal is detected.
Note 2: Set “0” when writing.
Note 3: If SUSPEND="1", set this bit to “1” (keep this bit set for a minimum of 1ms and a maximum of 15ms).

Figure 2.8.27. USB power management register

(2) USB Suspend Function

In the M30245 group, the USB suspend status flag (SUSPEND) of USB power management register (address 0282_{16}) is set to “1” when the suspend signal has been received from the host CPU (or not detected any bus activity on the D+/D- line for at least 3ms). Simultaneously, the USB suspend interrupt request occurs.

To shift into the USB suspend state, control the USB function control unit in the following procedure. Further, for changes in frequency synthesizer control register (address $03DC_{16}$) or system clock control register1 (address 0007_{16}), clearing the corresponding bit of protect register (address $000A_{16}$) is required.

● **USB Suspend Mode Control**

1: Set USB clock enable bit of USB control register to “0”. Do not write to USB internal registers (other than the USBC, USBAD, and frequency synthesizer related registers) when the USB clock has been disabled in the suspend state.

2: During the bus power supply operation as a low-power device, control it to reduce the total driving current to 500μA or below (or 2.5mA or below when remote wakeup has been enabled as a high-power device by the host CPU). For details of the power control in suspend, refer to USB2.0 specification.

3: Set frequency synthesizer enable bit of frequency synthesizer control register to “0”.

4: Set the return interrupt from the USB suspend state. Set USB resume interrupt control register (address 0058_{16}). (When remote wakeup has been enabled, enable interrupt control register of the peripheral functions used in remote wakeup.)

5: Set I flag to “1”.

6: Stop the system clock by setting all clock stop control bit (bit 0 of CM1) to 1.

7: During the bus power supply operation, after setting the low-power consumption mode by disabling interrupts that are not used in return from the USB suspend state, etc., execute the low-power consumption mode.

Note: When the device is in self-powered operation, the above control is not required.

(3) USB Resume Function

● **Returning Routine from USB Suspend State**

To return from the USB suspend state, the M30245 group uses the USB resume interrupt occurred by receiving the resume signal from the host or the interrupt for remote wakeup for transmitting the resume signal to the host.

- Returning by Resume Interrupt

When the resume signal is received from the host CPU during the USB suspend state (when detected any bus activity on D+/D- line in suspend detect state), the USB resume interrupt request occurs, setting “1” to interrupt request bit of USB resume interrupt control register (address 0058_{16}). When the USB clock is operated, the USB suspend status flag is automatically set to “0” at this time. For returning from the suspend state by the USB resume interrupt, follow the procedure below:

1: Return the USB function control unit. (Refer to the next page.)
2: Enable other functions as circumstances demand.

- Returning by Remote Wakeup

When clock operation is started by the remote wakeup interrupt (other than the USB resume interrupt) during the USB suspend state, transmit the resume signal to the host CPU as follows:

1: Return the USB function control unit. (Refer to the next page.)
2: Set USB remote wakeup bit to “1” and transmit the resume signal to the host CPU. (Retain “1” for min. 1ms to max. 15ms.)
3: Set USB remote wakeup bit to “0” and complete the resume signal transmission. The USB suspend status flag is automatically cleared at this time.

Also, when returning from the stop mode, the main clock dividing ratio has been set to 8-dividing mode, for which resetting is required. Wait for enough oscillation stabilization time before resetting main clock division select bit of system clock control register 0 (address 0006_{16}). (For details, refer to “Clock-Generating Circuit” of Chapter 1 “Hardware”.)

- Returning Routine of USB Function Control Unit

To return from the USB suspend state to the previous state, perform return control of the USB function control unit as follows.

Further, clearing the bit of protect register (address $000A_{16}$) is required for changes in frequency synthesizer control register (address $03DC_{16}$).

1: Set frequency synthesizer enable bit of frequency synthesizer control register to “1”.
2: Wait for 3ms.
3: Check that frequency synthesizer lock status bit of frequency synthesizer control register is set to “1”. When this bit has been set to “0”, wait for 0.1ms, and then, check again. Repeat the re-check until the bit becomes “1”.
4: Set USB clock enable bit of USB control register to “1”.

Do not write to USB-related registers (other than the USBC, USBAD, and frequency synthesizer related registers) when the USB clock has been disabled in the suspend state.

(4) USB Suspend Interrupt Request Processing Routine

When using the USB suspend interrupt, set USB suspend interrupt control register (address 0056$_{16}$).

In the USB suspend interrupt, when the USB suspend status flag (SUSPEND) of USB power management register is set to “1”, the interrupt request occurs.

The USB suspend interrupt request processing routine is shown in Figure 2.8.28 and Figure 2.8.29.

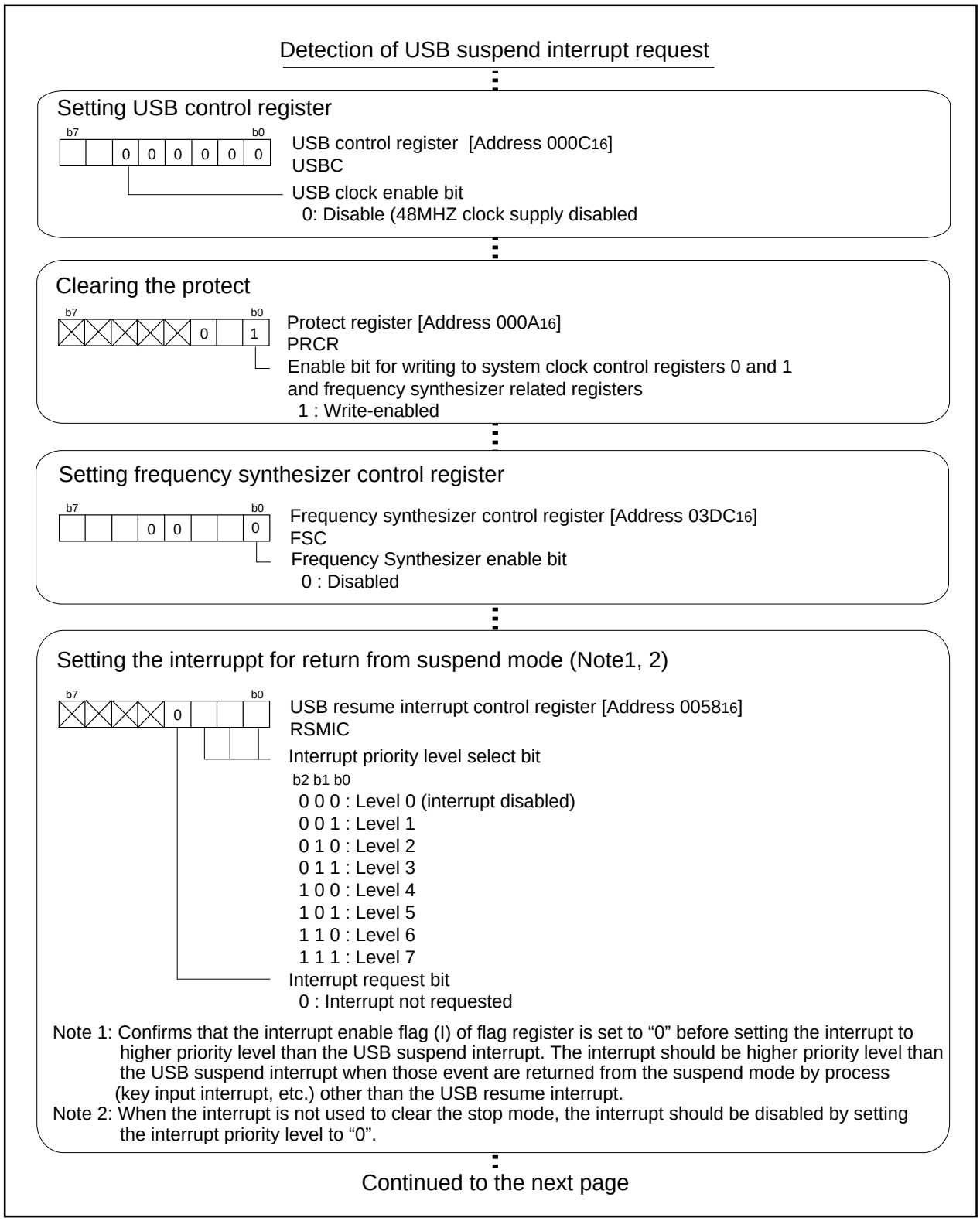

Detection of USB suspend interrupt request
Setting USB control register
b7 b0
0 0 0 0 0 0
USB control register [Address 000C16]
USBC
USB clock enable bit
0: Disable (48MHZ clock supply disabled
Clearing the protect
b7 b0
0 1
Protect register [Address 000A16]
PRCR
Enable bit for writing to system clock control registers 0 and 1
and frequency synthesizer related registers
1 : Write-enabled
Setting frequency synthesizer control register
b7 b0
0 0 0
Frequency synthesizer control register [Address 03DC16]
FSC
Frequency Synthesizer enable bit
0 : Disabled
Setting the interruppt for return from suspend mode (Note1, 2)
b7 b0
0
USB resume interrupt control register [Address 005816]
RSMIC
Interrupt priority level select bit
b2 b1 b0
0 0 0 : Level 0 (interrupt disabled)
0 0 1 : Level 1
0 1 0 : Level 2
0 1 1 : Level 3
1 0 0 : Level 4
1 0 1 : Level 5
1 1 0 : Level 6
1 1 1 : Level 7
Interrupt request bit
0 : Interrupt not requested
Note 1: Confirms that the interrupt enable flag (I) of flag register is set to “0” before setting the interrupt to higher priority level than the USB suspend interrupt. The interrupt should be higher priority level than the USB suspend interrupt when those event are returned from the suspend mode by process (key input interrupt, etc.) other than the USB resume interrupt.
Note 2: When the interrupt is not used to clear the stop mode, the interrupt should be disabled by setting the interrupt priority level to “0”.
Continued to the next page

Figure 2.8.28. USB suspend interrupt request processing routine (1)

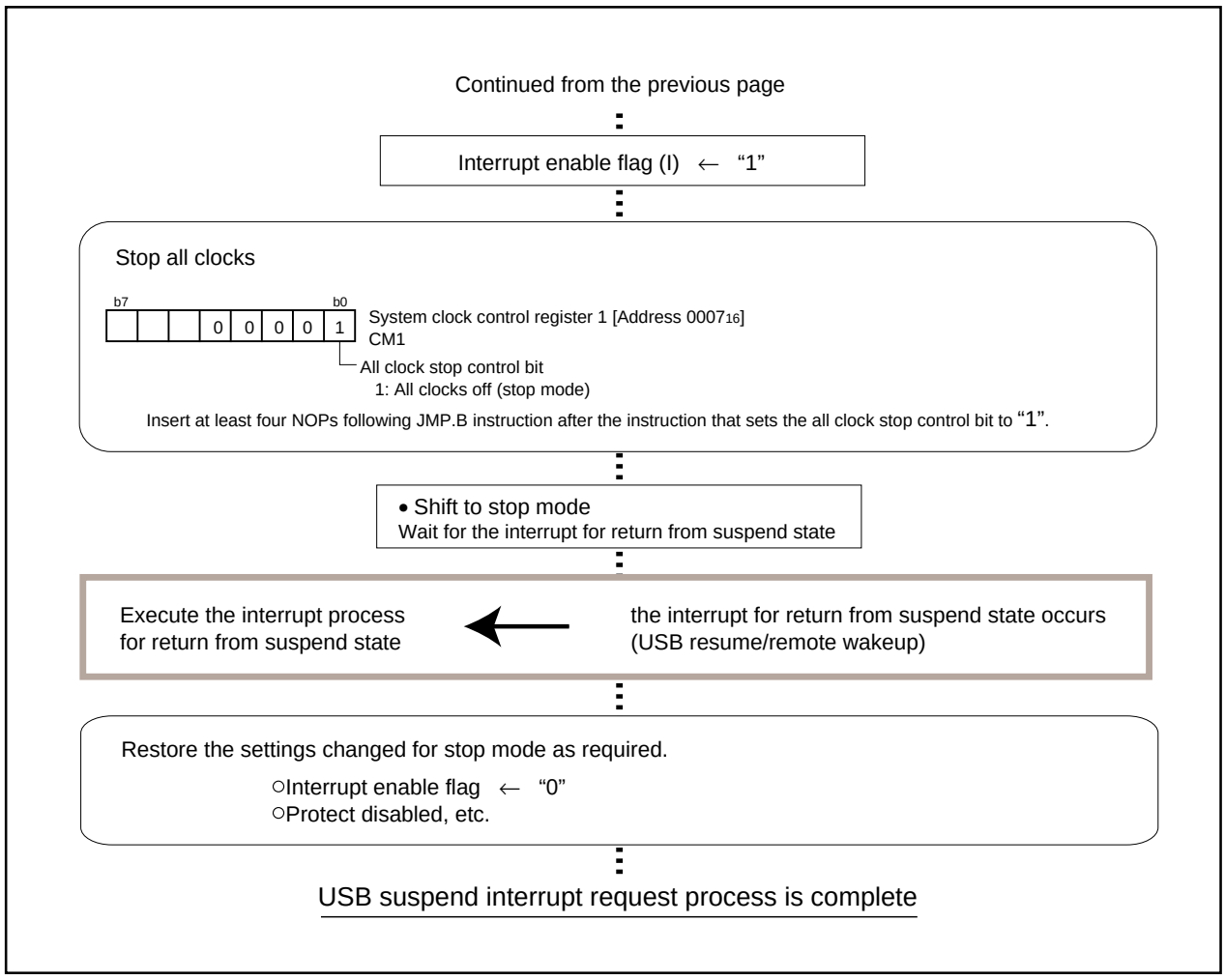

Continued from the previous page
Interrupt enable flag (I) ← "1"
Stop all clocks
b7
b0
0 0 0 0 1
System clock control register 1 [Address 0007₁₆]
CM1
All clock stop control bit
1: All clocks off (stop mode)
Insert at least four NOPs following JMP.B instruction after the instruction that sets the all clock stop control bit to "1".
• Shift to stop mode
Wait for the interrupt for return from suspend state
Execute the interrupt process for return from suspend state
the interrupt for return from suspend state occurs (USB resume/remote wakeup)
Restore the settings changed for stop mode as required.
○Interrupt enable flag ← "0"
○Protect disabled, etc.
USB suspend interrupt request process is complete

Figure 2.8.29. USB suspend interrupt request processing routine (2)

(5) USB Resume Interrupt Request Processing Routine

When the resume signal is received from the host CPU during the USB suspend mode (when detected any bus activity on D+/D- line in suspend detect state), the USB resume interrupt request occurs. The USB suspend status flag is automatically set to "0" at this time.

The USB resume interrupt request processing routine is shown in Figure 2.8.30.

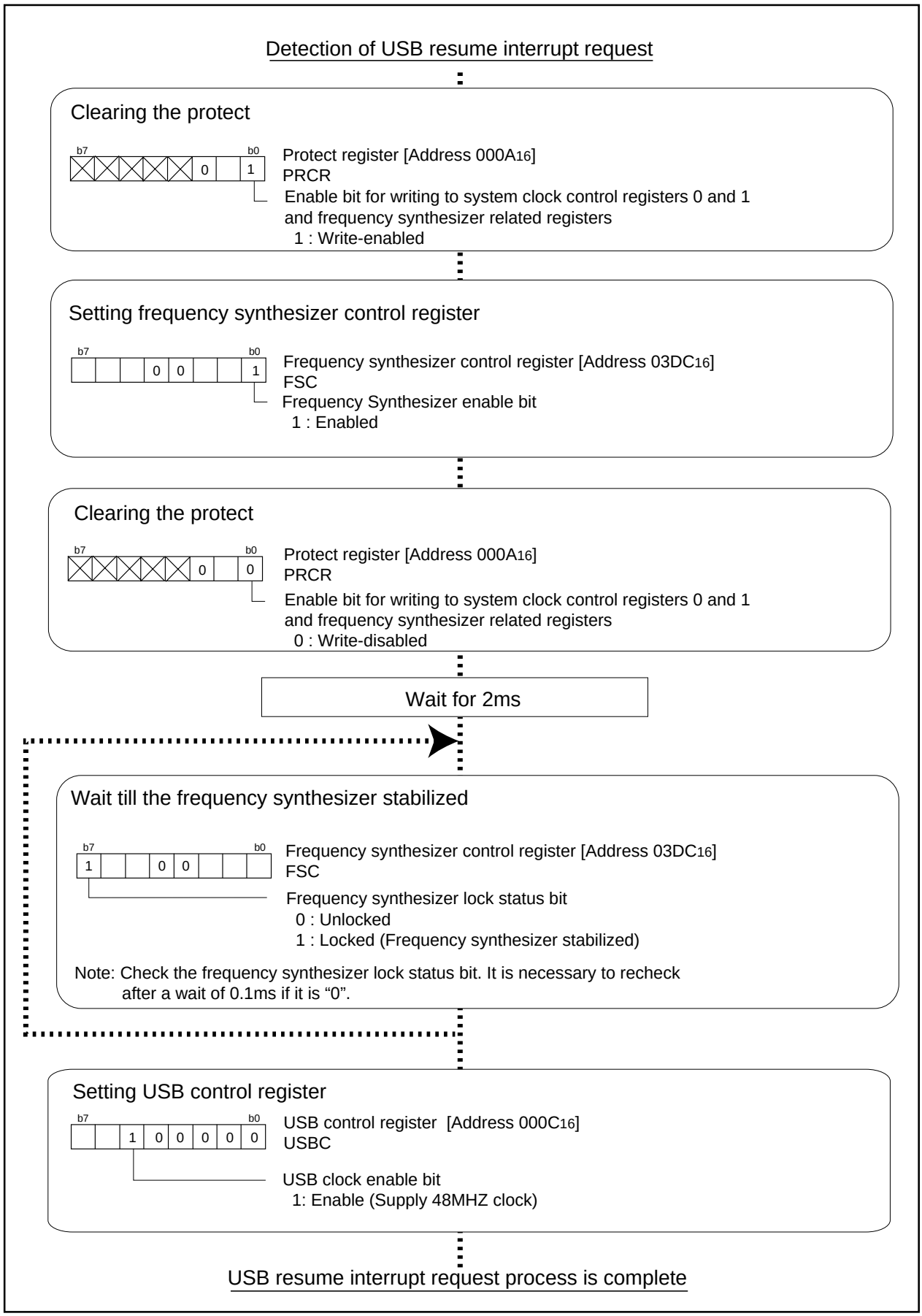

Detection of USB resume interrupt request
Clearing the protect
b7 b0
0 1
Protect register [Address 000A16]
PRCR
Enable bit for writing to system clock control registers 0 and 1
and frequency synthesizer related registers
1 : Write-enabled
Setting frequency synthesizer control register
b7 b0
0 0 1
Frequency synthesizer control register [Address 03DC16]
FSC
Frequency Synthesizer enable bit
1 : Enabled
Clearing the protect
b7 b0
0 0
Protect register [Address 000A16]
PRCR
Enable bit for writing to system clock control registers 0 and 1
and frequency synthesizer related registers
0 : Write-disabled
Wait for 2ms
Wait till the frequency synthesizer stabilized
b7 b0
1 0 0
Frequency synthesizer control register [Address 03DC16]
FSC
Frequency synthesizer lock status bit
0 : Unlocked
1 : Locked (Frequency synthesizer stabilized)
Note: Check the frequency synthesizer lock status bit. It is necessary to recheck
after a wait of 0.1ms if it is “0”.
Setting USB control register
b7 b0
1 0 0 0 0 0
USB control register [Address 000C16]
USBC
USB clock enable bit
1: Enable (Supply 48MHZ clock)
USB resume interrupt request process is complete

Figure 2.8.30. USB resume interrupt request processing routine

2.8.5 USB Operation (Endpoint 0)

Endpoint 0 is used only for control transfer.

Endpoint 0 FIFO consists of total 256 bytes including IN (transmit) FIFO and OUT (receive) FIFO each respectively of 128 bytes. The starting position is allocated from the 3072nd byte to the 3327th byte of the endpoint FIFO. Both the endpoint 0 FIFO size and the starting position are fixed. The FIFO size to be used is determined by the USB endpoint 0 maximum packet size.

The packet data received from the host CPU are written in endpoint 0 OUT FIFO. When a data receive request is issued by the host CPU while data already exists in the OUT FIFO, responds with NAK automatically. When packet data are transmitted to the host CPU, the transmit data are written in the endpoint 0 IN FIFO. When a data transmit request is issued by the host CPU before the data are written in IN FIFO, responds with NAK automatically. A packet data can realize higher transmit/receive by enabling continuous transfer.

When an error is detected in control transfer, responds with STALL automatically and the error detection is reported. Based on the status of the endpoint 0 communication, data transmit/receive is controlled in accordance with the device request which is received from the host CPU.

(1) Related Registers

● **USB address register**

USB address register maintains 7 bits addresses of the USB function control unit that are allocated by the host CPU. The USB function control unit of the M30245 group responds to the token packet for the address retained in this register.

When the USB function control unit is in the initial state or has received a reset signal from the host CPU, this register value is the default address “0000_{16}”. When the USB block has been disabled (bit 7 of USB control register is set to “0”), this register value is the address “0000_{16}”.

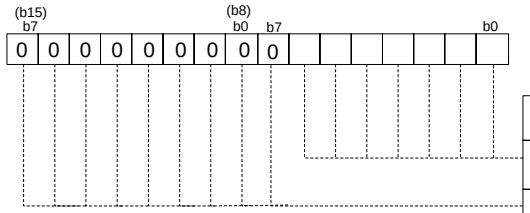
After receiving the SET_ADDRESS request from the host CPU, rewrite USB address register and update the address.

For rewriting USB address register, follow the procedure below:

- When the device is in the default state (USB address register value is “0000_{16}”):
  1: When USB address register has received the SET_ADDRESS request from the host CPU, store the new address data in the USB address register.
  2: When the status phase of the SET_ADDRESS request is completed , USB address register is automatically rewritten into the address written in above-mentioned 1:. When the status phase is not normally completed, USB address register is not rewritten.
- When the device is in the address state (USB address register value is other than “0000_{16}”):
  1: When USB address register has received the SET_ADDRESS request from the host CPU, confirm that the status phase of SET_ADDRESS request completes.
  2: Store the new address data in USB address register. USB address register is rewritten into the new address data.

The configuration of USB address register is shown in Figure 2.8.31.

USB Function Address register



Symbol: USBA | Address: 028016 | When reset: 000016

Bit Symbol	Bit Name	Function	R	W
FUNAD6-0	Function address	7-bit programmable function address	O	O
Reserved		Must always be “0”	O	O

Figure 2.8.31. USB address register

● **USB endpoint 0 control and status register**

USB endpoint 0 control and status register consists of the bits concerning control information and status information of endpoint 0.

• **OUT_BUF_RDY Flag**

This flag shows the status of OUT FIFO.

The OUT_BUF_RDY flag is set to “1” in the following cases:

- The setup packet is received.
- One data packet is received from the host CPU in the data phase.

Set the OUT_BUF_RDY flag to “0” by setting “1” to CLR_OUT_BUF_RDY bit after reading the OUT FIFO.

• **IN_BUF_RDY Flag**

This flag shows the status of IN FIFO.

When this flag is set to “1”, shows that data to be transmitted to the host CPU exists in FIFO.

When transmission of the IN FIFO data is completed or when the SETUP_END flag is set to “1”, this flag is cleared to “0”.

• **SETUP Flag**

When the setup packet is received from the host CPU, this flag is set to “1”.

The OUT_BUF_RDY flag is also set to “1” at this time.

This flag is cleared by setting “1” to CLR_SETUP bit.

• **DATA_END Flag**

This flag shows the status phase control of control transfer.

After the status phase is started or when a new SETUP packet is received, this flag is automatically set to “0”.

When DATA_END_MASK bit is set to “1” (at the time of resetting), the DATA_END flag is always set to “0”, and endpoint 0 interrupt factor does not occur by clearing the DATA_END flag to “0”.

• **FORCE_STALL Flag**

This flag shows the error occurrence in control transfer.

This flag is set to “1” for reporting an error when at least one of the following conditions occurs:

- IN token without SETUP stage is received.
- An incorrect data toggle is received in the STATUS stage. (DATA0 is used.)
- An incorrect data toggle is received in the SETUP stage. (DATA1 is used.)
- Data exceeding the one specified in the SETUP stage are required. (IN token is received after the DATA_END flag is set.)

- Data exceeding the one specified in the SETUP stage are required. (An OUT token is received after the DATA_END flag is set.)
- Data exceeding the one specified are received in USB endpoint 0 MAXP register.

Except for when an incorrect data toggle is received in the SETUP stage, on occurrence of the above condition, STALL is transmitted to the IN/OUT token with a problem. When an incorrect data toggle is received in the SETUP stage, ACK is returned to the SETUP stage and STALL is returned to next IN/OUT token.

The STALL handshake occurring by the above condition completes the control transfer in operation by being transmitted to one transaction. The packet after STALL handshake is regarded as the start of new control transfer. Set this flag to “0” by writing “1” to CLR_FORCE_STALL bit.

- **SETUP_END flag**

This flag shows the interrupt in control transfer.

This flag is set to “1” when at least one of the following conditions occurs:

- Transmission of the data which had amount of data set by setup phase during data phase processing is completed. (The status phase is started before the DATA_END flag is set.)
- A new SETUP packet is received before status phase is completed.

When this flag is set to “1” and transmit data to the host CPU exists, IN_BUF_RDY bit is cleared to “0” and the IN FIFO data is destroyed. Discontinue access to FIFO and process the preceding setup.

Also, when a new SETUP packet is received right after the SETUP_END flag is set to “1” (when the next new SETUP packet is received before the data phase or the status phase is completed), the SETUP flag and the OUT_BUT_RDY flag in addition to the SETUP_END flag are set to “1”, indicating that a new SETUP packet data exists in OUT FIFO.

Set the SETUP_END flag to “0” by writing “1” to CLR_SETUP_END bit.

- **CLR_OUT_BUF_RDY bit**

This bit controls clearing of the OUT_BUF_RDY flag.

This bit is set to “1” after reading the data packet from OUT FIFO. When this flag is written to “1”, the OUT_BUF_RDY flag is cleared to “0”.

When the OUT_BUF_RDY flag is set to “1” by receiving the SETUP token, the USB function control unit responds with NAK to the data request from the host CPU.

Until decoding of request data from the host CPU is completed, do not set this bit to “1” (nor the OUT_BUF_RDY flag is set to “0”.)

- **SET_IN_BUF_RDY bit**

This bit controls setting of the IN_BUF_RDY flag to “1”.

Completion of one buffer data write is notified to the USB function control unit.

Set this bit to “1” after writing the data packet to IN FIFO. When this bit is written to “1”, the IN_BUF_RDY flag is set to “1”.

- **CLR_SETUP bit**

This bit controls clearing of the SETUP flag.

Set this bit to “1” after decoding the SETUP packet. When this bit is written to “1”, the SETUP flag is cleared to “0”.

- **SET_DATA_END bit**

  This bit controls setting of the DATA_END flag to “1”.

  When the last data has been written in IN FIFO in the IN data phase or when the last data has been read from OUT FIFO in the OUT data phase, set this bit to “1”. When this bit is set to “1”, the DATA_END flag is set to “1”. At this time simultaneously, set the CLR_OUT_BUF_RDY bit or SET_IN_BUF_RDY bit to “1”.

  Completion of processing of the data which had amount of data set by setup phase is notified to the USB function control unit, and the process shifts into status phase processing.

- **CLR_FORCE_STALL bit**

  This bit controls clearing of the FORCE_STALL flag.

  When this bit is written to “1”, the FORCE_STALL flag is cleared to “0”.

- **SEND_STALL bit**

  This bit controls the STALL response to the host CPU.

  To responds with STALL when an invalid request has received from the host CPU, set this bit to “1” simultaneously as CLR_OUT_BUF_RDY bit is set to “1”.

  When this bit is set to “1”, the USB function control unit transmits, to the host CPU, the STALL handshake concerning all the IN/OUT transactions. When a new valid SETUP packet is received, clear a data by writing “0” in this bit.

- **DATA_END_MASK bit**

  This bit controls whether the DATA_END flag clear becomes an endpoint 0 interrupt factor.

  When this bit is set to “1”, clearing the DATA_END flag does not become an endpoint 0 interrupt factor. This bit is set to “1” at the time of reset.

The configuration of USB endpoint 0 control and status register is shown in Figure 2.8.32.

USB Endpoint 0 Control and Status register

Symbol: EP0CS
Address: 0298$_{16}$
When reset: 2000$_{16}$

Bit Symbol	Bit Name	Function	R	W
EP0CSR0	OUT_BUF_RDY flag	0 : No setup packet or data set ready for unload 1 : Setup packet or data set ready for unload	O	X
EP0CSR1	IN_BUF_RDY flag	0 : No data set ready for transmit 1 : Data set ready for transmit	O	X
EP0CSR2	SETUP flag	0 : No setup packet ready for unload 1 : Data set ready for transmit	O	X
EP0CSR3	DATA_END flag	0 : DATA_END not set by CPU or DATA-END is set by CPU then status phase starts 1 : DATA-END set by CPU	O	X
EP0CSR4	FORCE_STALL flag	0 : No protocol violation detected 1 : Protocol violation detected	O	X
EP0CSR5	SETUP_END flag	0 : No premature completion of control transfer 1 : Premature completion of control transfer	O	X
EP0CSR6	CLR_OUT_BUF_RDY	0 : No action 1 : Data set unloaded from the OUT buffer	O	O Note
EP0CSR7	SET_IN_BUF_RDY	0 : No action 1 : Data set loaded in IN buffer (sets IN_BUF_RDY flag)	O	O Note
EP0CSR8	CLR_SETUP	0 : No action 1 : Clears SETUP flag	O	O Note
EP0CSR9	SET_DATA_END	0 : No action 1 : Last data pcket transferred to/from buffer	O	O Note
EP0CSR10	CLR_FORCE_STALL	0 : No action 1 : Clears FORCE_STALL flag	O	O Note
EP0CSR11	CLR_SETUP_END	0 : No action 1 : Clears SETUP_END flag	O	O Note
EP0CSR12	SEND_STALL	0 : No EP0 STALL by CPU 1 : EP0 STALL by CPU	O	O
EP0CSR13	DATA_END_MASK	0 : Clearing DATA_END event causing EP0 interrupt is unmasked 1 : Clearing DATA_END event causing EP0 interrupt is masked	O	O
Reserved		Must always be “0”	O	O

Note: Always read a “0”.

Figure 2.8.32. USB endpoint 0 control and status register

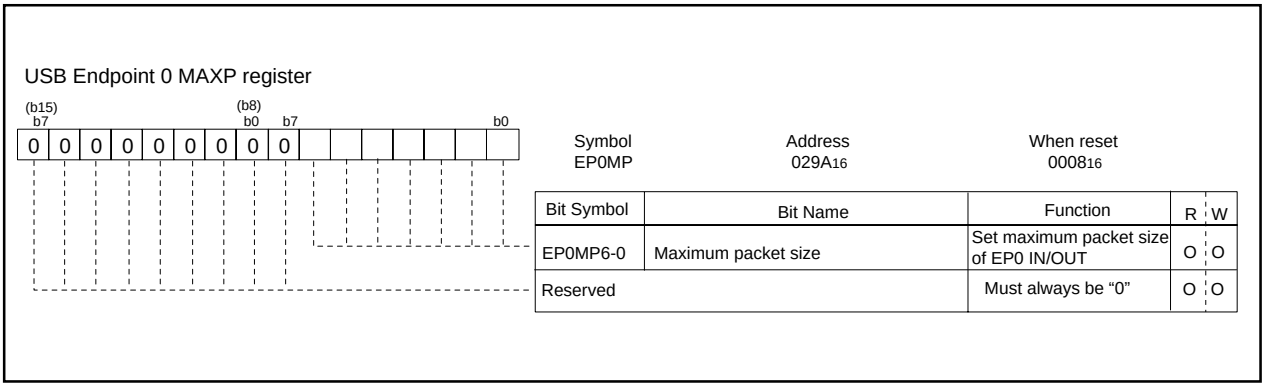
● **USB endpoint 0 MAXP register**

This register indicates the IN/OUT maximum packet size of endpoint 0.

When a GET_DESCRIPTION request is received from the host CPU, write to this register to change the IN/OUT maximum packet size value of endpoint 0.

Set the packet size value (8, 16 or 32 bytes) specified by control transfer. The default value is 8 bytes.

The configuration of USB endpoint 0 MAXP register is shown in Figure 2.8.33.



Symbol: EP0MP | Address: 029A16 | When reset: 000816

Bit Symbol	Bit Name	Function	R	W
EP0MP6-0	Maximum packet size	Set maximum packet size of EP0 IN/OUT	O	O
Reserved		Must always be “0”	O	O

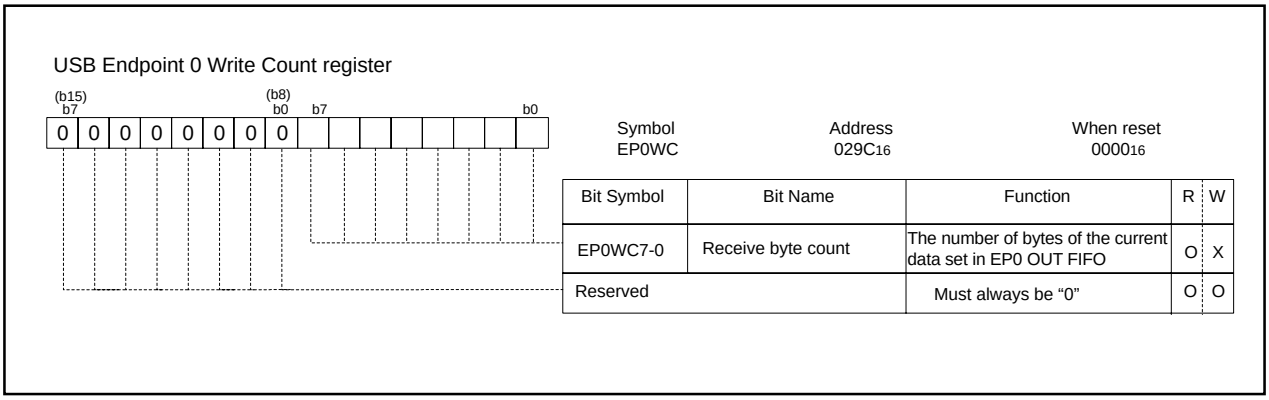
Figure 2.8.33. USB endpoint 0 MAXP register

● **USB endpoint 0 OUT write count register**

This register contains the number of bytes of the current data set in the OUT FIFO. When the USB function control unit completes the data packet receive from the host CPU, set the value of this register. When one buffer data receive completes, read this register and determine the number of bytes to be read from OUT FIFO. This register value is not decremented even if the data is read from OUT FIFO.

When CLR_OUT_BUF_RDY bit of the EP0CSR is set to “1”, this register value is cleared to “0”.

The configuration of USB endpoint 0 OUT write count register is shown in Figure 2.8.34.



Symbol: EP0WC | Address: 029C16 | When reset: 000016

Bit Symbol	Bit Name	Function	R	W
EP0WC7-0	Receive byte count	The number of bytes of the current data set in EP0 OUT FIFO	O	X
Reserved		Must always be “0”	O	O

Figure 2.8.34. USB endpoint 0 OUT write count register

(2) Control Transfer: Endpoint 0 Receive

The endpoint 0 receives the packet data from the host CPU in the setup stage or the data stage by the control write transfer. When the receive of a valid SETUP packet or a data packet completes, the SETUP flag and the OUT_BUF_RDY flag are automatically set to “1”, and the number of bytes of receive data are set in USB endpoint 0 OUT write count register (address $029C_{16}$). Read the data of only amount equal to received byte count from the endpoint 0 OUT FIFO. Every time that one-byte data is read from OUT FIFO, the internal write pointer is automatically decremented by “2” in word access and by “1” in byte access. The contents of internal write pointer cannot be read.

When the data read from OUT FIFO is completed, simultaneously set “1” to CLR_OUT_BUF_RDY bit and SET_DATA_END bit. (When the SETUP packet is received, clearing the SETUP flag by setting “1” to CLR_SETUP bit is required.) Therefore, the OUT_BUF_RDY flag is cleared and the DATA_END flag is set to “1”.

The USB function unit proceeds to the status phase process when the DATA_END flag is set to “1”. When the status phase completes, the DATA_END flag is cleared to “0”.

Manage the stage of control transfer by software.

When the SETUP packet is received, the USB endpoint 0 interrupt occurs regardless of setting of continuous transfer mode enable bit (The OUT_BUF_RDY flag and the SETUP flag are set to “1”).

● **Example of one packet data receive procedure**

1: Check that one packet data is received in OUT FIFO.

2: Read the number of bytes of receive packet data from USB endpoint 0 OUT write count register. Determine the amount of data to read from OUT FIFO.

3: Read the data of only amount equal to determined in the above-mentioned 2: from OUT FIFO. To analyze the received data, the subsequent stage and operation are determined based on the read data.

4: With CLR_OUT_BUF_RDY bit being set to “1”, the OUT_BUF_RDY flag is cleared to complete fetch of the receive one packet, and manage the next stage control. At this time, when the SETUP packet is received, the SETUP flag is also cleared by setting “1” to CLR_SETUP bit.

- For shifting into the status stage even if the next data to be received or transmitted does not exist, simultaneously set CLR_OUT_BUF_RDYbit (and also CLR_SETUP bit for the SETUP packet) and SET_DATA_END bit to “1”.
- For responding with STALL response to the next token, simultaneously set CLR_OUT_BUF_RDY bit (and also CLR_SETUP bit for the SETUP packet) and SEND_STALL bit to “1”.
- When the valid new SETUP packet is received after setting SEND_STALL bit, clear SEND_STALL bit and set CLR_OUT_BUF_RDY bit to “1” (and also CLR_SETUP bit for the SETUP packet).

(3) Control Transfer: Endpoint 0 Transmit

The endpoint 0 transmits the packet data to the host CPU in the data stage by the control read after completion of receive request analysis process in the setup stage.

Write one packet data to be transmitted in IN FIFO. Every time that one-byte data is written in IN FIFO, the internal write pointer is automatically incremented by “2” in word access and by “1” in byte access. The contents of internal write pointer cannot be read. When the data write in IN FIFO is completed, set IN_BUF_RDY flag to “1” by setting “1” to SET_IN_BUF_RDY bit. When an empty packet (with 0 data length) is transmitted, data is not written in IN FIFO and SET_IN_BUF_RDY bit is set to “1”.

At this time, one packet transmission is prepared and is transmitted by the USB function control unit in the next IN token.

The IN_BUF_RDY flag is automatically set to “0”, when one packet data transmission is completed to the host CPU (or on receiving ACK) or when the SETUP_END flag is set to “1”. After writing the last data packet in IN FIFO, set SET_IN_BUF_RDY bit to “1” and, simultaneously set SET_DATA_END bit to “1”. Both the IN_BUF_RDY flag and DATA_END flag are set to “1”. When the DATA_END flag becomes “1” after completion of transmission of the last data packet, the USB function control unit proceeds to the status phase processing.

When the status phase is completed, the DATA_END flag is cleared to “0”.

Manage the stage of control transfer by software.

● **Example of one packet data transmit procedure**

1: Check that the packet data does not exist in IN FIFO (the IN_BUF_RDY flag is “0”) before writing the data of the 2nd packet and after of data stage.

2: The data is written in IN FIFO based on the amount specified on the SETUP stage.
When an empty packet (with 0 data length) is transmitted, the data is not written in IN FIFO.
The subsequent stage and operation are determined.

3: With SET_IN_BUF_RDY bit being set to “1”, one packet transmission is prepared, and the next stage control is managed.

- For shifting into the status stage even if the next data to be transmitted does not exist, simultaneously set SET_IN_BUF_RDY bit and SET_DATA_END bit.
- When the next empty packet is transmitted, set SET_IN_BUF_RDY bit to “1” and continue transmitting processing.

(4) Control Transfer: Example of Standard Device Request Receive

The control transfer includes the setup stage, data stage and status stage.

Which one of write transfer, read transfer and no data transfer is executed in the data stage is determined by the content of the setup data acquired in the setup stage.

Examples of the receive processing routine of the SET_ADDRESS request and the GET_CONFIGURATION request are described.

For rewriting USB address register when the SET_ADDRESS request is received, follow the procedure below:

● **When the device is in the default state (USB address register value is “0”):**

1: When USB address register is received the SET_ADDRESS request from the host CPU, store the new address data in the USB address register.

2: When the status phase of the SET_ADDRESS request is completed, USB address register is rewritten into the address written in above-mentioned 1:. When the status phase is not normally completed, USB address register is not rewritten.

● **When the device is in the address state (USB address register value is other than “0”):**

1: When USB address registeris received the SET_ADDRESS request from the host CPU, confirm that the status phase of SET_ADDRESS request completes.

2: Store the new address data in USB address register.

The USB function control unit applies this address to all the subsequent device accesses.

The SET_ADDRESS request is shown in Figure 2.8.35, the device address acquisition processing routine of USB SET_ADDRESS request is shown in Figure 2.8.36 and Figure 2.8.37, the device configuration notification processing routine of GET_CONFIGURATION request is shown in Figure 2.8.38 and Figure 2.8.39.

Describe these processing to endpoint 0 interrupt processing.

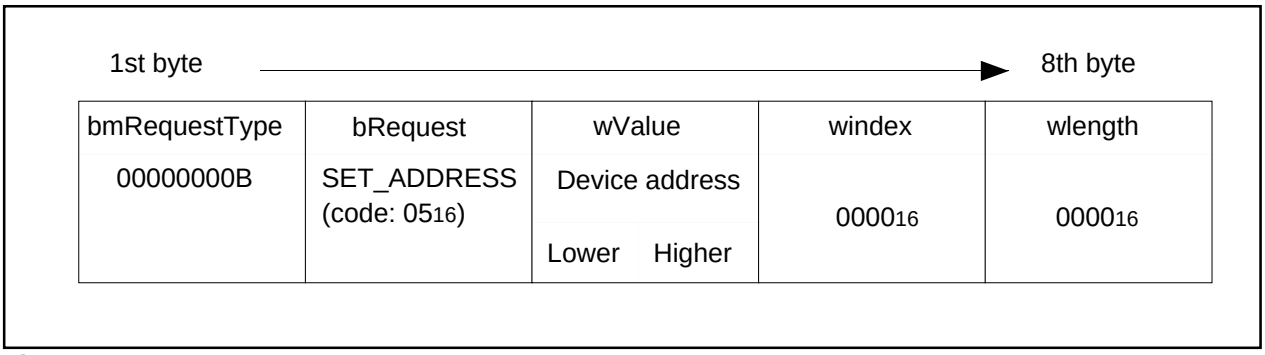


Figure 2.8.35. SET_ADDRESS Request

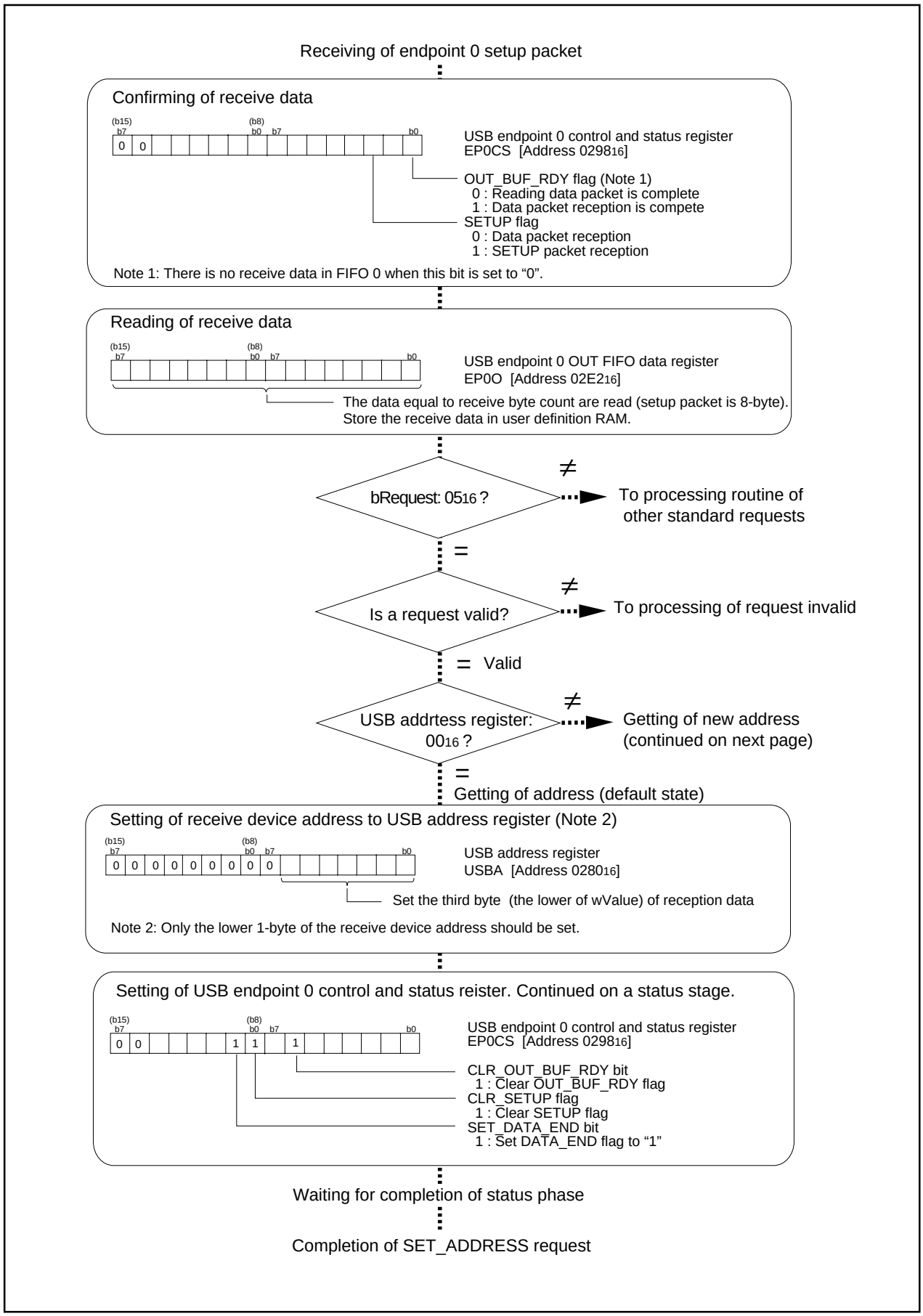

Receiving of endpoint 0 setup packet
Confirming of receive data
(b15) b7 (b8) b0 b7 b0
0 0
USB endpoint 0 control and status register
EP0CS [Address 029816]
OUT_BUF_RDY flag (Note 1)
0 : Reading data packet is complete
1 : Data packet reception is compete
SETUP flag
0 : Data packet reception
1 : SETUP packet reception
Note 1: There is no receive data in FIFO 0 when this bit is set to "0".
Reading of receive data
(b15) b7 (b8) b0 b7 b0
USB endpoint 0 OUT FIFO data register
EP0O [Address 02E216]
The data equal to receive byte count are read (setup packet is 8-byte).
Store the receive data in user definition RAM.
bRequest: 0516 ?
≠
To processing routine of other standard requests
=
Is a request valid?
≠
To processing of request invalid
= Valid
USB addrtess register: 0016 ?
≠
Getting of new address (continued on next page)
=
Getting of address (default state)
Setting of receive device address to USB address register (Note 2)
(b15) b7 (b8) b0 b7 b0
0 0 0 0 0 0 0 0
USB address register
USBA [Address 028016]
Set the third byte (the lower of wValue) of reception data
Note 2: Only the lower 1-byte of the receive device address should be set.
Setting of USB endpoint 0 control and status reister. Continued on a status stage.
(b15) b7 (b8) b0 b7 b0
0 0 1 1 1
USB endpoint 0 control and status register
EP0CS [Address 029816]
CLR_OUT_BUF_RDY bit
1 : Clear OUT_BUF_RDY flag
CLR_SETUP flag
1 : Clear SETUP flag
SET_DATA_END bit
1 : Set DATA_END flag to "1"
Waiting for completion of status phase
Completion of SET_ADDRESS request

Figure 2.8.36. Processing routine (1) for getting device address when receiving SET_ADDRESS request

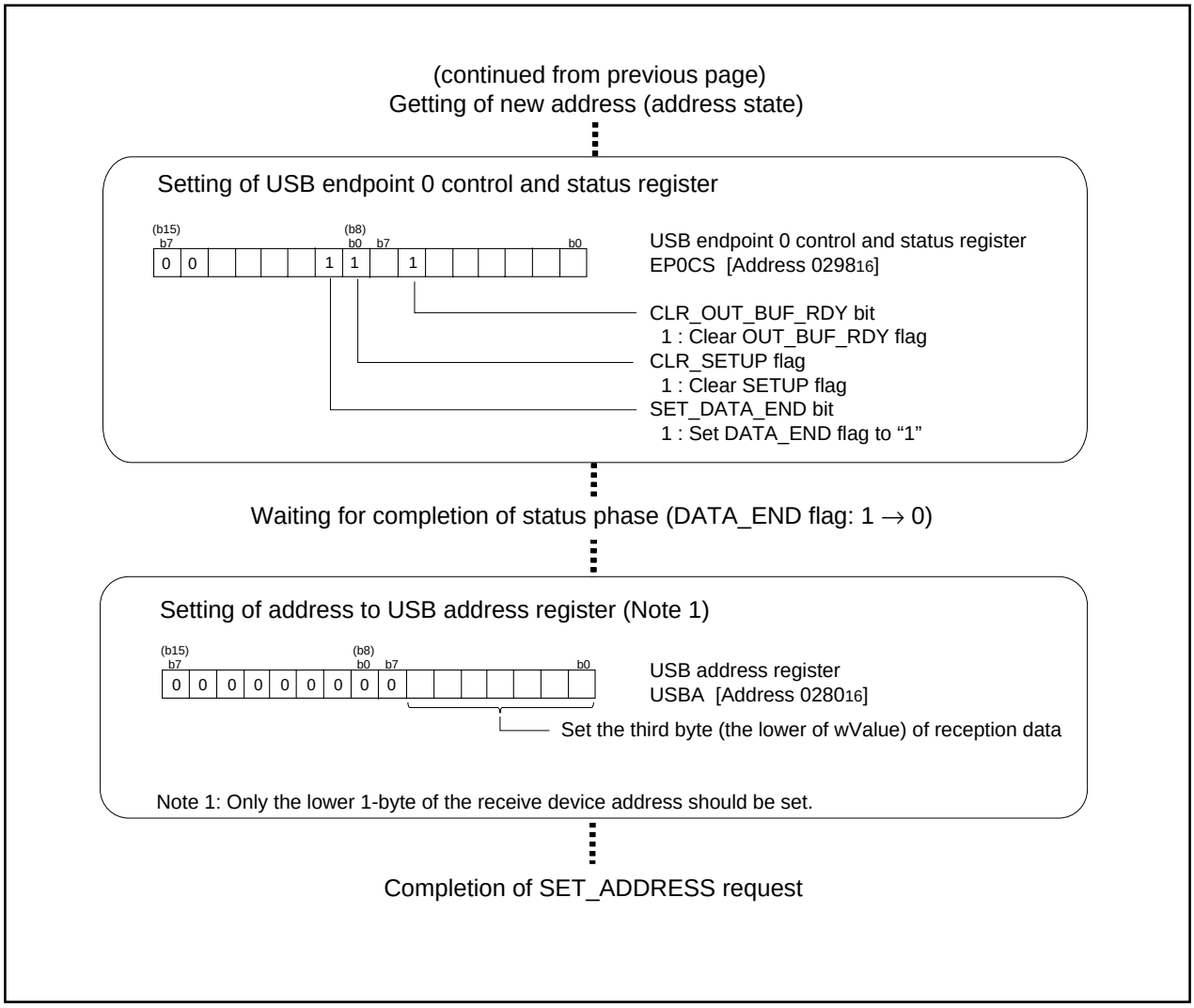

(continued from previous page)
Getting of new address (address state)
Setting of USB endpoint 0 control and status register
(b15)
b7
(b8)
b0
b7
b0
0 0 1 1 1
USB endpoint 0 control and status register
EP0CS [Address 029816]
CLR_OUT_BUF_RDY bit
1 : Clear OUT_BUF_RDY flag
CLR_SETUP flag
1 : Clear SETUP flag
SET_DATA_END bit
1 : Set DATA_END flag to "1"
Waiting for completion of status phase (DATA_END flag: 1 → 0)
Setting of address to USB address register (Note 1)
(b15)
b7
(b8)
b0
b7
b0
0 0 0 0 0 0 0 0 0
USB address register
USBA [Address 028016]
Set the third byte (the lower of wValue) of reception data
Note 1: Only the lower 1-byte of the receive device address should be set.
Completion of SET_ADDRESS request

Figure 2.8.37. Processing routine (2) for getting device address when receiving SET_ADDRESS request

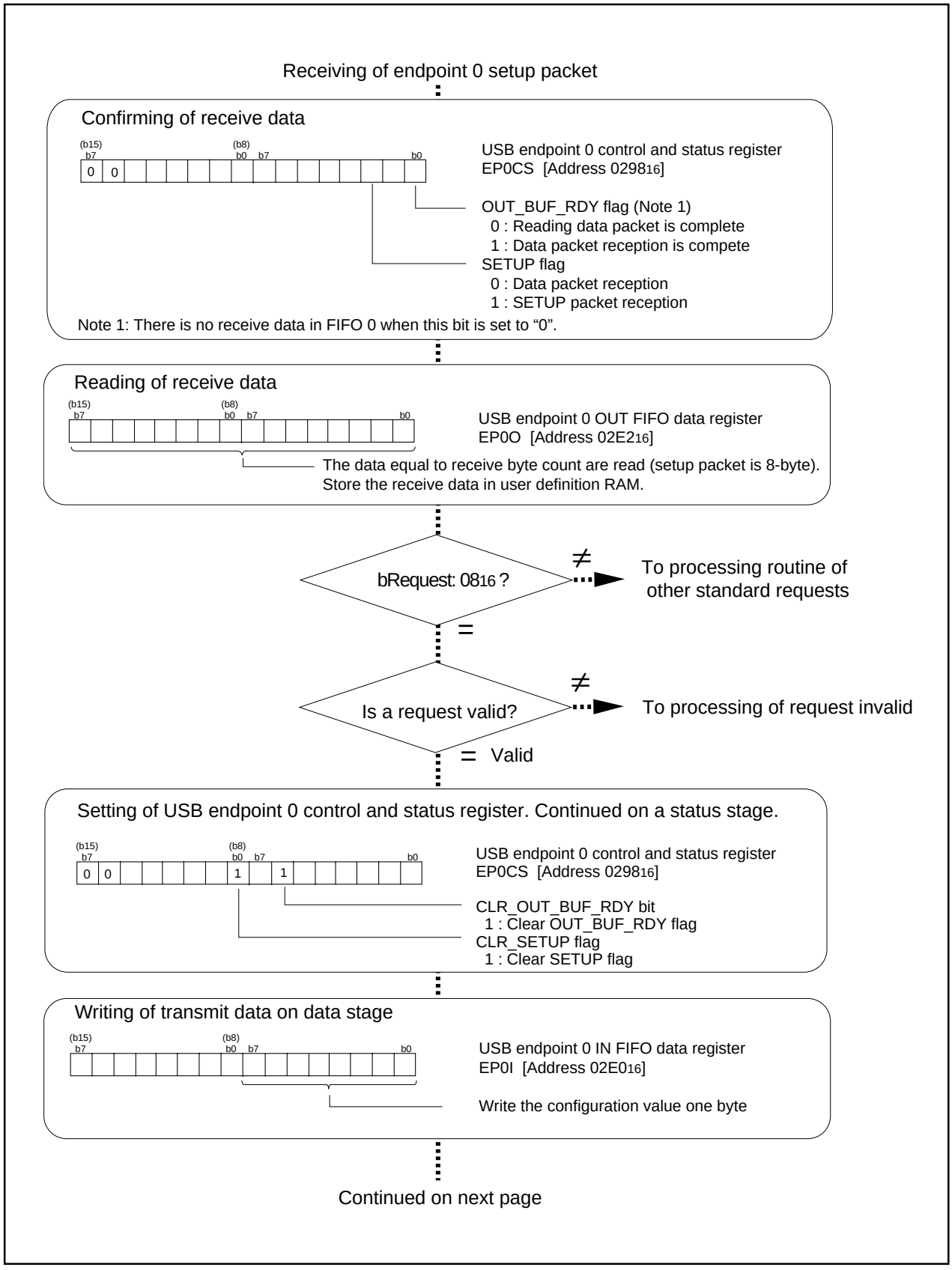

Receiving of endpoint 0 setup packet
Confirming of receive data
(b15) b7
(b8) b0 b7
b0
0 0
USB endpoint 0 control and status register
EP0CS [Address 029816]
OUT_BUF_RDY flag (Note 1)
0 : Reading data packet is complete
1 : Data packet reception is compete
SETUP flag
0 : Data packet reception
1 : SETUP packet reception
Note 1: There is no receive data in FIFO 0 when this bit is set to "0".
Reading of receive data
USB endpoint 0 OUT FIFO data register
EP0O [Address 02E216]
The data equal to receive byte count are read (setup packet is 8-byte).
Store the receive data in user definition RAM.
bRequest: 0816 ?
≠
To processing routine of other standard requests
=
Is a request valid?
≠
To processing of request invalid
= Valid
Setting of USB endpoint 0 control and status register. Continued on a status stage.
USB endpoint 0 control and status register
EP0CS [Address 029816]
CLR_OUT_BUF_RDY bit
1 : Clear OUT_BUF_RDY flag
CLR_SETUP flag
1 : Clear SETUP flag
Writing of transmit data on data stage
USB endpoint 0 IN FIFO data register
EP0I [Address 02E016]
Write the configuration value one byte
Continued on next page

Figure 2.8.38. Device configuration notification processing routine (1) when receiving GET_CONFIGURATION request

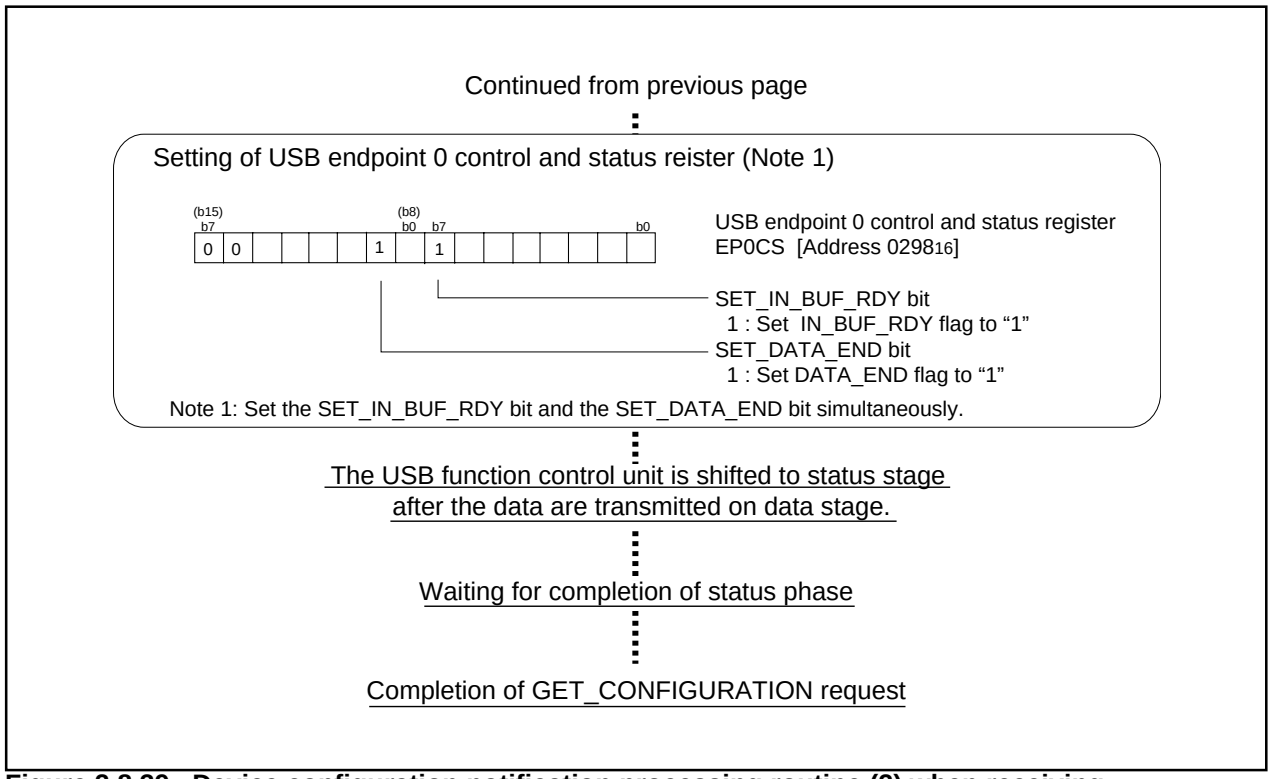

Continued from previous page
Setting of USB endpoint 0 control and status reister (Note 1)
(b15)
b7
(b8)
b0
b7
b0
0
0
1
1
USB endpoint 0 control and status register
EP0CS [Address 029816]
SET_IN_BUF_RDY bit
1 : Set IN_BUF_RDY flag to “1”
SET_DATA_END bit
1 : Set DATA_END flag to “1”
Note 1: Set the SET_IN_BUF_RDY bit and the SET_DATA_END bit simultaneously.
The USB function control unit is shifted to status stage
after the data are transmitted on data stage.
Waiting for completion of status phase
Completion of GET_CONFIGURATION request

Figure 2.8.39. Device configuration notification processing routine (2) when receiving GET_CONFIGURATION request

2.8.6 USB Operation (Endpoints 1 to 4 Receive)

Endpoints 1 to 4 can apply to the isochronous transfer, bulk transfer and interrupt transfer.
The endpoints 1 to 4 respectively have their IN (transmit) FIFOs and OUT (receive) FIFOs.
For using the endpoints 1 to 4 OUT, enable each endpoint OUT FIFO by USB endpoint enable register (address $028E_{16}$). The size and the starting location (every 64 bytes) of each endpoint x(x=1 to 4) OUT FIFO can be set according to the user's system. The buffer size of OUT FIFO can be set to a maximum of 1024 bytes per 64 bytes for one endpoint. When the double buffer mode is enabled, the buffer which has twice as much as the set size is available for the OUT FIFO. The size and starting location of FIFO, the double buffer mode enable can be set by USB endpoint x OUT FIFO configuration register (EPxOFC).
When one buffer data is received from the host CPU, the data are written to the endpoint x OUT FIFO and the number of bytes of receive packet data are stored in USB endpoint x OUT write count register. When a data receive request from the host CPU occurs while data are already written and OUT FIFO cannot be received, NAK is automatically transmitted in bulk transfer/interrupt transfer, and an overrun occurs in isochronous transfer, not receiving the packet data.
The data receive from the host CPU is controlled based on the communication status of endpoints 1 to 4 OUT.
The default of endpoints 1 to 4 is bulk transfer. Each endpoint should be initialized in order to use other transfer modes.
The receive of endpoints 1 to 4 can select the following functions:

Continuous Receive Mode

This function is used for receiving data from the host PC at a higher speed. This mode can be set only for endpoints 1 to 4 OUT bulk transfer. With continuous transfer mode bit of the EPxOFC being set to “1”, the continuous receive mode is enabled. The USB function control unit writes the receive data from the host PC in OUT FIFO sequentially by the maximum packet size that is set in USB endpoint x OUT MAXP register (EPxOMP). (When the last one packet is smaller than the size set in the EPxOMP, it is received as a short packet.)
When continuous receive mode is enabled, the buffer size has to be equal to an integral multiple of the EPxOMP. Further, the user's system has to be comprehended beforehand that the receive data from the host PC are equal to the buffer size or includes a short packet.

AUTO_CLR Function

When receive data from OUT FIFO are read, both the OUT_BUF_STS0 and the OUT_BUF_STS1 flags are updated without CLR_OUT_BUF_RDY bit being set to “1”. The AUTO_CLR function is enabled by setting AUTO_CLR bit of the EPxOCS to “1”. The AUTO_CLR function is available both in the continuous receive mode and in the continuous receive mode disable of endpoints 1 to 4 OUT (Not available with endpoint 0).

(1) Related Registers

● **USB endpoint x(x=1 to 4) OUT control and status register**

•OUT_BUF_STS1, OUT_BUF_STS0 flags

These flags indicate OUT FIFO state.

At the time of reading the receive data from the host PC, read these flags to confirm the OUT FIFO state. When the OUT_BUF_STS1 and the OUT_BUF_STS0 flags are respectively set to "00_2", there are no data in OUT FIFO. When they are respectively set to "10_2", there are only one buffer data in double buffer. (Invalid for single buffer.) When they are respectively set to "11_2", there are one buffer data in single buffer while there are two buffer data in double buffer. When they are respectively set to "01_2", it is invalid.

These flags are updated when one of the following events occurs:

- One valid buffer data is successfully received from the host.
- One buffer data is successfully fetched from OUT FIFO.
  CLR_OUT_BUF_RDY bit is set to "1" after read of one receive data from OUT FIFO completes. (When the AUTO_CLR function is enabled, these flags are updated without CLR_OUT_BUF_RDY bit being set to "1".)
- The OUT FIFO buffer data are flushed. (When FLUSH bit is set to "1".)

•OVER_RUN flag

This flag indicates occurrence of an overrun in isochronous transfer. The bit is valid only in isochronous transfer. When OUT FIFO is not empty and disables receiving at start of the OUT token from the host CPU, occurrence of an overrun is recognized, setting this bit to "1".

Clear this flag by writing "1" to CLR_OVER_RUN bit.

•FORCE_STALL flag

This flag indicates occurrence of a packet size error.

When the data packet, which size exceeds USB endpoint x OUT MAXP register value, is transmitted from the host CPU, this flag becomes "1". While this bit is set to "1", the USB function control unit does not receive packet data. If it is in bulk transfer, also, STALL handshake is transmitted to the host CPU.

Clear this flag by writing "1" to CLR_FORCE_STALL bit.

•DATA_ERR flag

This flag indicates occurrence of data error in isochronous transfer. The bit is valid only in isochronous transfer. If any bit stuffing error or CRC error is detected in the received packet, this flag becomes "1".

Clear this flag by writing "1" to CLR_DATA_ERR bit.

•CLR_OUT_BUF_RDY bit

This bit controls OUT FIFO. Set this bit to "1" after one receive buffer data is read from OUT FIFO. Completion of one buffer data fetch is notified to the USB function control unit and, simultaneously, the OUT_BUF_STS0 and OUT_BUF_STS1 flags are updated.

When the AUTO_CLR function is enabled, this bit does not need to be set up.

•CLR_OVER_RUN bit

The OVER_RUN flag is cleared to "0" by setting "1" to this bit.

•CLR_FORCE_STALL bit

The FORCE_STALL flag is cleared to "0" by setting "1" to this bit.

•CLR_DATA_ERR bit

The DATA_ERR flag is cleared to "0" by setting "1" to this bit.

•TOGGLE_INIT bit

This bit initializes data toggle sequence bit in bulk/interrupt transfer.

With this bit being set to “1”, the PID of the next packet to be received from the host CPU becomes DATA0. When initialization of the data toggle sequence is requested from the host CPU at the time of configuration, etc., set TOGGLE_INIT bit and initialize PID to DATA0 before starting the OUT end-point communication.

At this time, the internal read/write counter of OUT FIFO is also initialized.

On completing PID initialization, this bit is automatically cleared to “0”.

•FLUSH bit

This bit controls the OUT FIFO packet.

With this bit being set to “1”, one buffer data received in OUT FIFO is flushed out from the OUT FIFO.

- When there is one buffer data in OUT FIFO, the OUT FIFO becomes empty.
  At this time, the OUT_BUF_STS1 and OUT_BUF_STS0 flags are updated from “11_2”(“10_2”) to “00_2”.
- When there are two buffer data in OUT FIFO, the older data is flushed out from the OUT FIFO.
  At this time, the OUT_BUF_STS1 and OUT_BUF_STS0 flags are updated from “11_2” to “10_2” (This indicates that one more buffer data is left inside the OUT FIFO).

The receive data may be destroyed if this bit is set to “1” during USB transfer.

Read the OUT_BUF_STS1 and OUT_BUF_STS0 flags and confirm that there are data in the OUT FIFO, and then, set this bit to “1”.

On completing one buffer data destruction, this bit is automatically cleared to “0”.

•ISO bit

Set this bit to “1” in order to use an endpoint in isochronous transfer. Set this bit to “0” in order to use an endpoint in bulk/interrupt transfer.

•SEND_STALL bit

This bit controls the STALL response to the host CPU in bulk transfer/interrupt transfer.

Set this bit to “1” when the OUT endpoint is in STALL state. While this bit is set to “1”, the USB function control unit transmits the STALL handshake concerning all the OUT transactions to the host CPU. When the OUT endpoint has returned from STALL state, clear this bit by writing “0”. The OUT endpoint communication is resumed.

•AUTO_CLR bit

This bit controls setting of CLR_OUT_BUF_RDY bit.

With this bit being set to “1”, when one receive buffer data is read from OUT FIFO, the OUT_BUF_STS1 and OUT_BUF_STS0 flags are automatically updated without CLR_OUT_BUF_RDY bitbeing set to “1”. With this bit being set to “0”, on completing one receive buffer data fetch from OUT FIFO, CLR_OUT_BUF_RDY bit has to be set to “1” by software.

The configuration of USB endpoint x OUT control and status register is shown in Figure 2.8.40.

USB Endpoint x OUT Control and Status register

Symbol: EPxOCS (x = 1 - 4)

Address: 02B6$_{16}$, 02BE$_{16}$, 02C6$_{16}$, 02CE$_{16}$

When reset: 0000$_{16}$

Bit Symbol	Bit Name	Function	R	W
OUTxCSR0	OUT_BUF_STS0 flag	These two bits indicate the EPx OUT buffer status: Bit1 Bit0 0 0 : No data set in the OUT buffer 0 1 : Single buffer mode: N/A Double buffer mode: N/A 1 0 : Single buffer mode: N/A Double buffer mode: one data set in the OUT buffer 1 1 : Single buffer mode: one data set in the OUT buffer Double buffer mode: two data sets in the OUT buffer	O	X
OUTxCSR1	OUT_BUF_STS1 flag		O	X
OUTxCSR2	OVER-RUN flag	0 : No over run detected 1 : Over run detected	O	X
OUTxCSR3	FORCE_STALL flag	0 : No packet size larger than MAXP violation detected 1 : Packet size larger than MAXP violation detected	O	X
OUTxCSR4	DATA_ERR flag	0 : No data error detected 1 : Data error detected	O	X
OUTxCSR5	CLR_OUT_BUF_RDY	0 : No action 1 : Data set unloaded from the OUT buffer (updates status flags)	O	O Note
OUTxCSR6	CLR_OVER_RUN	0 : No action 1 : Clears OVER_RUN flag	O	O Note
OUTxCSR7	CLR_FORCE_STALL	0 : No action 1 : Clears FORCE_STALL flag	O	O Note
OUTxCSR8	CLR_DATA_ERR	0 : No action 1 : Clears DATA_ERR flag	O	O Note
OUTxCSR9	TOGGLE_INIT	0 : No action 1 : Initialize the next data PID as a DATA0 for reception	O	O Note
OUTxCSR10	FLUSH	0 : No action 1 : Flush out one data set	O	O Note
OUTxCSR11	ISO	0 : Select non-isochronous endpoint 1 : Select isochronous endpoint	O	O
OUTxCSR12	SEND_STALL	0 : No STALL by CPU 1 : STALL by CPU	O	O
OUTxCSR13	AUTO_CLR	0 : AUTO_CLR disabled 1 : AUTO_CLR enabled	O	O
Reserved		Must always be set to “0”	O	O

Note: Always read a “0” when writing to this bit.

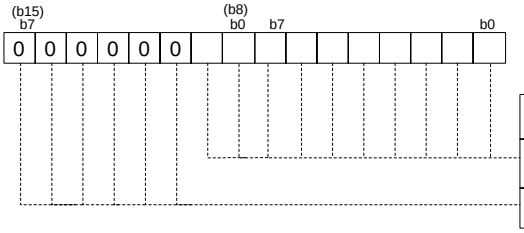
Figure 2.8.40. USB endpoint x(x=1 to 4) OUT control and status register

● **USB endpoint x(x=1 to 4) OUT MAXP register**

This register indicates endpoint x(x=1~4) OUT maximum packet size. The default value is 0 byte. When the endpoint is initialized due to any reason such as that the request for setting the endpoint (SET_DESCRIPTOR, SET_CONFIGURATION, SET_INTERFACE, etc.) is received from the host CPU, change the endpoint x OUT maximum packet size value by writing in this register. Set a packet size value specified for every transfer type to be used.

The configuration of USB endpoint x(x=1 to 4) OUT MAXP register is shown in Figure 2.8.41.

USB Endpoint x OUT MAXP register



Symbol	Address	When reset
EPxOMP (x = 1 - 4)	02B8$_{16}$, 02C0$_{16}$, 02C8$_{16}$, 02D0$_{16}$	0000$_{16}$

Bit Symbol	Bit Name	Function	R	W
OMAXP9-0	Maximum packet size	Set maximum packet size of EPx OUT	O	O
Reserved		Must always be “0”	O	O

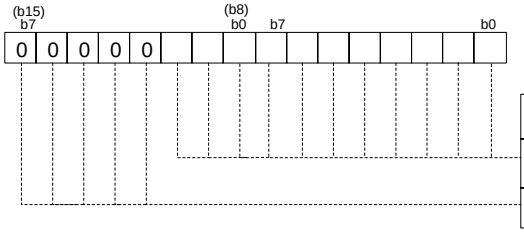
Figure 2.8.41. USB endpoint x(x=1 to 4) OUT MAXP register

● **USB endpoint x(x=1 to 4) OUT write count register**

This 11-bit register contains the number of bytes of one buffer data written in the endpoint x(x=1~4) OUT FIFO. This register is for read-only. When the USB function control unit completes the data packet receive from the host CPU, set the value of this register. When one buffer data receive completes, read this register and determine the byte count of the data to be read from OUT FIFO. This register value is not decremented even if the data are read from USB endpoint x OUT FIFO register . When this register is read while there are two buffer data in OUT FIFO in the double buffer mode, the number of bytes of the packet data received at first is already stored. When CLR_OUT_BUF_RDY bit is set to “1” after one buffer data is read from OUT FIFO, this register value is updated to the number of bytes of buffer data subsequently received.

The configuration of USB endpoint x(x=1 to 4) OUT write count register is shown in Figure 2.8.42.

USB Endpoint x OUT Write Count register



Symbol	Address	When reset
EPxWC (x = 1 - 4)	02BA$_{16}$, 02C2$_{16}$, 02CA$_{16}$, 02D2$_{16}$	0000$_{16}$

Bit Symbol	Bit Name	Function	R	W
WCNT10-0	Receive byte count	The byte count of receive one buffer data in the EPx OUT FIFO is set.	O	X
Reserved		Must always be “0”	O	O

Figure 2.8.42. USB endpoint x(x=1 to 4) OUT write count register

● **USB endpoint x(x=1 to 4) OUT FIFO configuration register**

This register sets endpoint x(x=1~4) OUT FIFO.

•BUF_NUM

This bit sets the starting location of the endpoint x(x= 1~4) OUT FIFO per 64 bytes. For example, when OUT FIFO is allocated, starting at the 320th byte, the set value is “0001012”.

•BUF_SIZ

This bit sets one buffer size of the endpoint x(x= 1~4) OUT FIFO per 64 bytes. For example, when 256 bytes is set, the set value is “01002”.

•DBL_BUF

With this bit being set to “1”, OUT FIFO of the corresponding endpoint is changed into double buffer mode. The byte count for a valid OUT FIFO becomes twice as much as the value specified by the BUF_SIZ at the time of double buffer. Set carefully not to overlap with the FIFO start position of other endpoints.

•CONTINUE

This bit enables continuous transfer mode.

Set this bit to “1” when continuous transfer is enabled. The bit is valid only in bulk transfer.

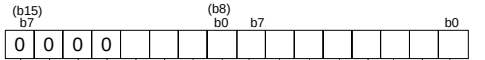
The USB function control unit writes the receive data from the host PC in OUT FIFO sequentially by one packet size (the maximum packet size set in the EPxOMP) and receives continuously until one buffer full or a short packet is received.

When continuous receive mode is enabled, the BUF_SIZ has to be equal to an integral multiple of the EPxOMP. Further, the user's system has to be comprehended beforehand that the receive data from the host PC are equal to the buffer size or includes a short packet.

Pay attention to the following when setting the BUF_NUM/BUF_SIZ:

- Not exceed 3072 bytes in OUT FIFO starting location + OUT FIFO size.
- Not overlap Endpoint FIFOs each other.

USB Endpoint x OUT FIFO register



Symbol: EPxOFC (x = 1 - 4)

Address: 02BC16, 02C416, 02CC16, 02D416

When reset: 000016

Bit Symbol	Bit Name	Function	R	W
BUF_NUM	FIFO buffer start number	Select the starting number for the EPx OUT FIFO (in units of 64 bytes) 000000 : buffer stating location = 0 000001 : buffer stating location = 64 000010 : buffer stating location = 128 ..... 101111 : buffer stating location = 3008 (last starting number)	O	O
BUF_SIZ	FIFO buffer size	Select the buffer size for the EPx OUT FIFO (in units of 64 bytes) 0000 : buffer stating location = 64 0001 : buffer stating location = 128 0010 : buffer stating location = 192 ..... 1111 : buffer stating location = 1024 (largest buffer size)	O	O
DBL_BUF	Double buffer mode	0 : Disabled 1 : Enabled	O	O
CONTINUE	Continuous transfer mode	0 : Disabled (Note) 1 : Enabled	O	O
Reserved		Must always be “0”	O	O

Note: Valid for bulk transfer type only

Figure 2.8.43. USB endpoint x(x=1 to 4) OUT FIFO configuration register

(2) Bulk Transfer: Endpoints 1 to 4 Receive

● **Setting of Transfer Type**

When endpoints 1 to 4 OUT are used for bulk transfer, ISO bit of USB endpoint x(x=1 to 4) OUT control and status register is set to “0” for bulk transfer setting.

Also, for initialization of toggle sequence bit in bulk transfer, set TOGGLE_INIT bit to “1” and initialize PID to DATA0.

Set by using USB endpoint x OUT FIFO configuration register in order to enable double buffer mode and continuous receive mode.

Set AUTO_CLR bit of USB endpoint x OUT control and status register to “1” in order to use the AUTO_CLR function.

● **Receive Operation**

When one packet data (Note 1) is received in OUT FIFO, the OUT_BUF_STS1 and the OUT_BUF_STS0 flags of the corresponding EPxOCS are automatically updated. In single buffer mode (when double buffer mode bit is “0”), these flags are updated from “00_2” to “11_2”. In double buffer mode, they are updated as follows:

- When the first one packet data (Note 1) of the double buffer has been written to the OUT FIFO and the second packet data (Note 1) is ready to be written, the OUT_BUF_STS1 and OUT_BUF_STS0 flags are updated from “00_2” to “10_2”.
- When two packet data (Note 1) have been written in OUT FIFO, the OUT_BUF_STS1 and OUT_BUF_STS0 flags are updated from “10_2” to “11_2”".

Note 1: In continuous transfer enable, read the description by substituting the underlined part with “buffer data”. The USB function control unit writes the receive data from the host PC in OUT FIFO sequentially by one packet size (the maximum packet size set in the EPxOMP), receives continuously until one buffer full or a short packet is received.

When the OUT token is received from the host CPU while SEND_STALL bit is set to “1”, STALL response is automatically returned.

When there is a packet space in OUT FIFO, on receiving the OUT token from the host CPU in the current data toggle sequence bit, the data are received and ACK response is returned. At this time, the OUT FIFO status is updated (updates the OUT_BUF_STS1 and OUT_BUF_STS0 flags) and data toggle sequence bit is toggled (DATA0 → DATA1 or DATA1 → DATA0). Further, the endpoint x OUT interrupt request occurs.

When there is a packet space in OUT FIFO, on receiving the OUT token from the host CPU in the toggle which is different from the current data toggle sequence bit, it is regarded that the ACK having responded for the packet previously received has dropped and, therefore, the host CPU has transmitted the same data. Only ACK response, therefore, is returned without receiving the data.

When the OUT token is received from the host CPU while there are already data in OUT FIFO and packet data cannot be received, NAK response is automatically returned.

When a packet, which size exceeds the maximum packet size, is transmitted from the host CPU, STALL response automatically is returned without receiving the data. At this time, the FORCE_STALL flag is set to “1” and, when error interrupt has been enabled by USB function interrupt enable register, an error interrupt request occurs (INTST8 is set to “1”).

When an error is detected in bulk OUT transfer, a response is not returned without ACK and NAK responses (Error checks such as CRC check and bit-justification, conforming to USB2.0 specification, are automatically performed. So the error does not have to be controlled by software).

● **Fetch of Receive Data**

On receiving one packet data (Note 2), the received packet data (Note 2) from OUT FIFO is read. Fetch one packet data (Note 2) in the following procedure:

1: Confirm that there are receive data in the OUT FIFO by the statuses of the OUT_BUF_STS1 and OUT_BUF_STS0 flags.

2: Determine the data byte count to be read from the OUT FIFO by reading USB endpoint x(x=1 to 4) OUT write count register .

3: Read the data byte count determined in the above 2: from the OUT FIFO.
Every time that 1(2)-byte data are read from the OUT FIFO, the internal write pointer is automatically decremented by one(two). (Content of the internal write pointer cannot be read.)

4: Set CLR_OUT_BUF_RDY bit to “1” to complete one receive packet data fetch (Note 2).
At this time, the OUT FIFO status (OUT_BUF_STS1 and OUT_BUF_STS0 flags) are updated, enabling a receive of the next one packet data (Note 2).

•In Single Buffer Mode
The OUT_BUF_STS1 and OUT_BUT_STS0 flags are updated from “11_2” (the OUT FIFO full) to “00_2” (the OUT FIFO empty).

•In Double Buffer Mode
When there are one more packet data (Note 2) in the OUT FIFO, the OUT_BUF_STS1 and OUT_BUF_STS1 flags are updated from “11_2” (the OUT FIFO full) to “10_2” (one data set in the OUT FIFO). In this case, the second packet data (Note 2) can be continuously fetched.
When there are no data packet does in the OUT FIFO, the OUT_BUF_STS1 and OUT_BUF_STS1 flags are updated from “10_2” (one data set in the OUT FIFO) to “00_2” (the OUT FIFO empty).

When one packet data (Note 2) is read from the OUT FIFO while the AUTO_CLR function is enabled (AUTO_CLR bit is “1”), the OUT_BUF_STS1 and OUT_BUF_STS0 flags are automatically updated without CLR_OUT_BUF_RDY bit being set to “1”.

Note 2: In continuous transfer enable, read the description by substituting the underlined part with “buffer data”. On receiving one buffer full (data equal to byte count set in the BUF_SIZ) or a short packet, one buffer data receive is completed. Also, the BUF_SIZ has to be equal to an integral multiple of the EPxOMP.

(3) Isochronous Transfer: Endpoints 1 to 4 Receive

● **Setting of Transfer Type**

When endpoints 1 to 4 OUT are used for isochronous transfer, ISO bit of USB endpoint x(x=1 to 4) OUT control and status register is set to “1” for isochronous transfer setting.

● **Receive Operation**

When there is a packet space in OUT FIFO, on receiving the OUT token from the host CPU, the data are received. At this time, the OUT FIFO status is updated, the endpoint x OUT interrupt request occurs. When an error is detected in the received packet, simultaneously, the DATA_ERR flag is set to “1”. (Error checks such as CRC check, conforming to USB2.0 specification, are automatically performed.)

When the OUT token is received from the host CPU while there are already data in OUT FIFO and packet data cannot be received, an overrun error occurs. At this time, the OVER_RUN flag is set to “1”.

Further, when a packet, which size exceeds the maximum packet size, is transmitted from the host CPU, the FORCE_STALL flag is set to “1” without receiving the data. While error interrupt has been enabled by USB function interrupt enable register, an error interrupt request occurs when any one of the OVER_RUN flag, FORCE_STALL flag or DATA_ERR flag is set to “1” (INTST8 is set to “1”).

● **Fetch of Receive Data**

The fetch procedure of endpoint x OUT receive data in the isochronous transfer is same as the bulk transfer.

Refer to “● Fetch of Receive Data” of “(2) Bulk Transfer: Endpoints 1 to 4 Receive”. (Although continuous transfer is valid for the bulk transfer only.)

(4) Interrupt Transfer: Endpoints 1 to 4 Receive

● **Setting of Transfer Type**

When endpoints 1 to 4 OUT are used for interrupt transfer, ISO bit of USB endpoint x(x=1 to 4) OUT control and status register is set to “0” for interrupt transfer setting.

Also, for initialization of toggle sequence bit in interrupt transfer, set TOGGLE_INIT bit to “1” and initialize PID to DATA0.

● **Receive Operation**

The endpoint x OUT receive operation in the interrupt transfer is same as the bulk transfer.

Refer to “● Receive Operation” of “(2) Bulk Transfer: Endpoints 1 to 4 Receive”.

● **Fetch of Receive Data**

The fetch procedure of endpoint x OUT receive data in the interrupt transfer is same as the bulk transfer.

Refer to “● Fetch of Receive Data” of “(2) Bulk Transfer: Endpoints 1 to 4 Receive”. (Although continuous transfer is valid for the bulk transfer only.)

(5) Precautions for Receive

● **Read from OUT FIFO**

Be sure to confirm the OUT_BUF_STS1 and OUT_BUF_STS0 flags states when reading data from the OUT FIFO. Based on these flags states, judge whether there are receive data in the OUT FIFO. Be sure to read the byte count of data specified by USB endpoint x OUT write count register value before setting CLR_OUT_BUF_RDYbitto “1” when reading data from the OUT FIFO. If the CLR_OUT_BUF_RDY bit is set to “1” during fetching of data from the OUT FIFO, the setting can cause malfunction of the internal read pointer.

Table 2.8.3. Status on Endpoint 1 to 4 OUT FIFOs

OUT_BUF_STS1	OUT_BUF_STS0	Single buffer [Specify OUT FIFO size by the BUF_SIZ*1]	Double buffer [OUT FIFO size = (The number of bytes specified by the BUF_SIZ*1)× 2]
0	0	No data Space equal to one buffer	No data Space equal to two buffer
0	1	Invalid	Invalid
1	0	Invalid	One data set in the OUT FIFO Space equal to one buffer
1	1	One data set in the OUT FIFO No space in the OUT FIFO	Two data set in the OUT FIFO No space in the OUT FIFO

*1: Bits 6 to 9 of EPxOFC.

● **PID Initialization**

When TOGGLE_INIT bit is set to “1”, the read/write counter inside the FIFO is initialized. To initialize the PID, set TOGGLE_INIT bit to “1” when the OUT FIFO is empty (the OUT_BUF_STS0 and OUT_BUF_STS1 flags are “00_2”).

(6) USB Receive (Endpoints 1 to 4 OUT): Example

The endpoints 1 to 4 OUT packet fetching routine (in continuous transfer disable) is shown in Figure 2.8.44. In addition to packet fetch process, error flag (OVER_RUN, FORCE_STALL, DATA_ERR) process is required for every transfer type.

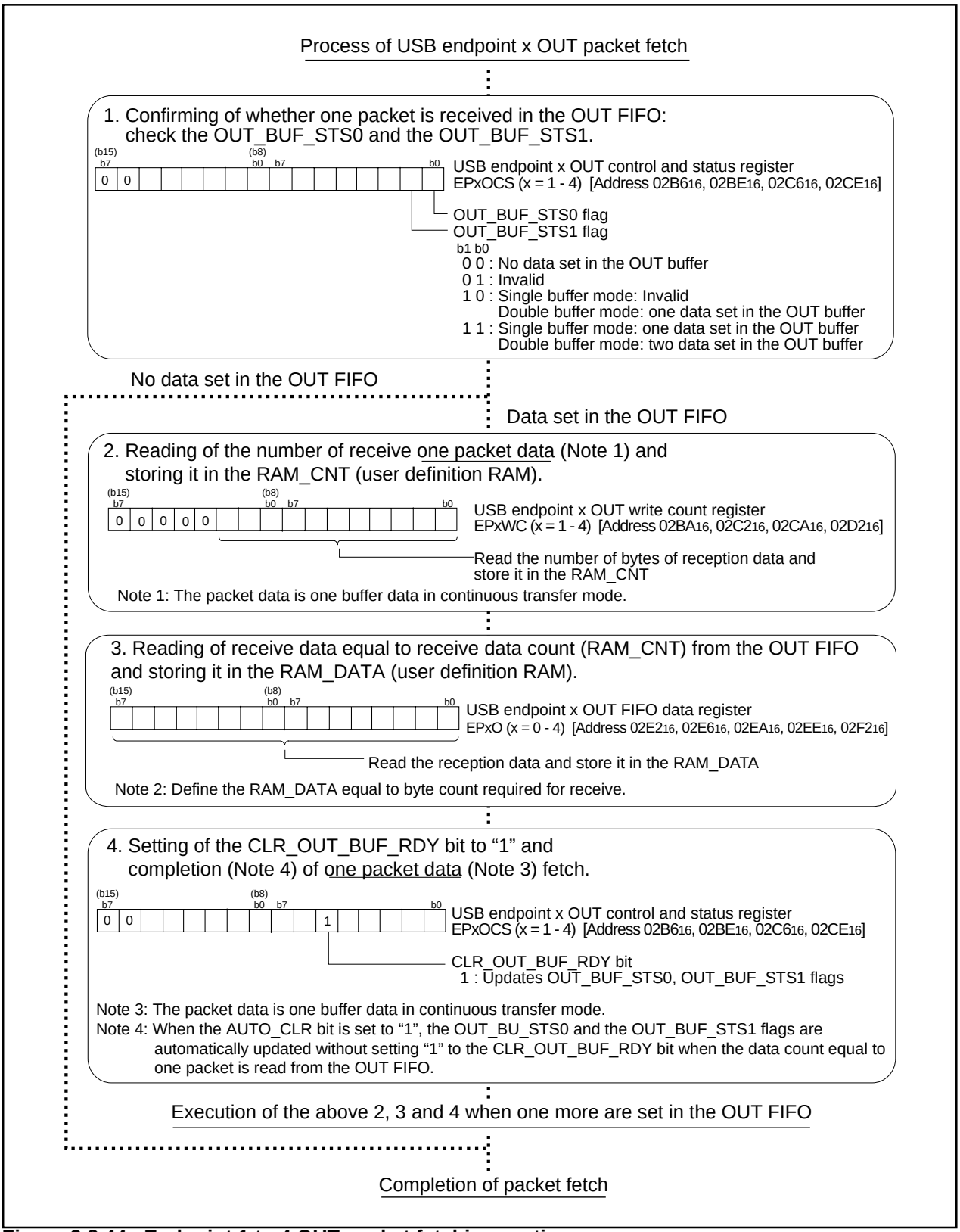

Process of USB endpoint x OUT packet fetch
1. Confirming of whether one packet is received in the OUT FIFO:
check the OUT_BUF_STS0 and the OUT_BUF_STS1.
(b15) (b8) b7 b0 b7 b0
0 0
USB endpoint x OUT control and status register
EPxOCS (x = 1 - 4) [Address 02B616, 02BE16, 02C616, 02CE16]
OUT_BUF_STS0 flag
OUT_BUF_STS1 flag
b1 b0
0 0 : No data set in the OUT buffer
0 1 : Invalid
1 0 : Single buffer mode: Invalid
Double buffer mode: one data set in the OUT buffer
1 1 : Single buffer mode: one data set in the OUT buffer
Double buffer mode: two data set in the OUT buffer
No data set in the OUT FIFO
Data set in the OUT FIFO
2. Reading of the number of receive one packet data (Note 1) and
storing it in the RAM_CNT (user definition RAM).
(b15) (b8) b7 b0 b7 b0
0 0 0 0 0
USB endpoint x OUT write count register
EPxWC (x = 1 - 4) [Address 02BA16, 02C216, 02CA16, 02D216]
Read the number of bytes of reception data and
store it in the RAM_CNT
Note 1: The packet data is one buffer data in continuous transfer mode.
3. Reading of receive data equal to receive data count (RAM_CNT) from the OUT FIFO
and storing it in the RAM_DATA (user definition RAM).
(b15) (b8) b7 b0 b7 b0
USB endpoint x OUT FIFO data register
EPxO (x = 0 - 4) [Address 02E216, 02E616, 02EA16, 02EE16, 02F216]
Read the reception data and store it in the RAM_DATA
Note 2: Define the RAM_DATA equal to byte count required for receive.
4. Setting of the CLR_OUT_BUF_RDY bit to “1” and
completion (Note 4) of one packet data (Note 3) fetch.
(b15) (b8) b7 b0 b7 b0
0 0 1
USB endpoint x OUT control and status register
EPxOCS (x = 1 - 4) [Address 02B616, 02BE16, 02C616, 02CE16]
CLR_OUT_BUF_RDY bit
1 : Updates OUT_BUF_STS0, OUT_BUF_STS1 flags
Note 3: The packet data is one buffer data in continuous transfer mode.
Note 4: When the AUTO_CLR bit is set to “1”, the OUT_BU_STS0 and the OUT_BUF_STS1 flags are
automatically updated without setting “1” to the CLR_OUT_BUF_RDY bit when the data count equal to
one packet is read from the OUT FIFO.
Execution of the above 2, 3 and 4 when one more are set in the OUT FIFO
Completion of packet fetch

Figure 2.8.44. Endpoint 1 to 4 OUT packet fetching routine

2.8.7 USB Operation (Endpoints 1 to 4 Transmit)

Endpoints 1 to 4 can apply to the isochronous transfer, bulk transfer and interrupt transfer.
The endpoints 1 to 4 respectively have their IN (transmit) FIFOs and OUT (receive) FIFOs.
For using the endpoints 1 to 4 IN, enable each endpoint IN FIFO by USB endpoint enable register (address $028E_{16}$). The size and the starting location (every 64 bytes) of each endpoint x(x=1~4) IN FIFO can be set according to the user's system. The buffer size of IN FIFO can be set to a maximum of 1024 bytes per 64 bytes for one endpoint. When the double buffer mode is enabled, the buffer which has twice as much as the set size is available for the IN FIFO. The size and starting location of FIFO, the double buffer mode enable can be set by USB endpoint x IN FIFO configuration register(EPxIFC).
When packet data are transmitted to the host CPU, the data are written to the endpoint x IN FIFO. When a data transmit request from the host CPU occurs before the data are written to IN FIFO, NAK is automatically transmitted in bulk transfer and interrupt transfer, or an empty packet (with 0 data length) is automatically transmitted in isochronous transfer.
The data transmitted to the host CPU is controlled based on the communication status of endpoints 1 to 4 IN. The default of endpoints 1 to 4 is bulk transfer. Each endpoint should be initialized in order to use other transfer modes.

The transmit of endpoints 1 to 4 can select the following functions.

Continuous Transmit Mode

This function is used for transmitting data at a higher speed. This mode can be set only for endpoints 1 to 4 IN bulk transfer. With continuous tranfer mode bit of the EPxIFC being set to “1”, the continuous transmit mode is enabled.
In continuous transmit mode, the USB function control unit, by dividing the data in IN FIFO into one packet size (the maximum packet size set in the EPxIMP) units, transmits them one by one to the host PC (When the last one packet is smaller than the size set in the EPxIMP, it is transmitted as a short packet.) When continuous transmit mode is enabled, the IN FIFO size has to be equal to an integral multiple of the maximum packet size.

AUTO_SET Function

With AUTO_SET bit of EPxICS being set to “1”, the AUTO_SET function is enabled. When transmit data of the buffer size (specified in the BUF_SIZ) is written to the IN FIFO in AUTO_SET enable state, the IN_BUF_STS0 and IN_BUF_STS1 flags are updated without SET_IN_BUF_RDY bit being set to “1”. However, when a short packet (data whose size is smaller than the EPxIMP value in continuous transfer disable or than the BUF_SIZ value in continuous transfer enable) has been written, the IN_BUF_STS1, IN_BUF_STS0 flags are not automatically updated. In these cases, the completion of data transmit ready is indicated by setting SET_IN_BUF_RDY bit to “1”. The AUTO_SET function is useable both in continuous transmit mode and in continuous transmit mode disable of endpoints 1 to 4 IN (Not available with endpoint 0).

(1) Related Registers

● **USB ISO control register**

This register controls isochronous transfer of endpoints 1 to 4. This register setting is valid for all the isochronous transfer IN endpoints that are used simultaneously.

•AUTO_FLUSH bit

This bit controls transmit packet data destruction in isochronous transfer. This bit can be used only when setting “1” to both ISO_UPDATE bit and ISO bit. The bit is valid only for the IN endpoints 1 to 4 in isochronous transfer.

While ISO_UPDATE bit =“1”, AUTO_FLUSH bit =“1”, and ISO bit (INxCSR8)=“1”, the USB function control unit, at the time of detecting SOF packet (on receiving from the host PC or on the artificial SOF operation), automatically flushes old data packet inside the IN FIFO if both the IN_BUF_STS1 and the IN_BUF_STS0 flags are “1” (IN FIFO full state). In isochronous transfer for double buffer, use the AUTO FLUSH function.

•ISO_UPDATE bit

This bit controls the transmit timing of packet data in isochronous transfer. The bit is valid only for the IN endpoints 1 to 4 in isochronous transfer.

While ISO_UPDATE bit =“0” and ISO bit =“1”, the USB function control unit, at the time of receiving of an IN token from the host CPU, transmits one packet of the IN FIFO data if SET_IN_BUF_RDY bit of the corresponding endpoint has been set beforehand to “1”.

While ISO_UPDATE bit =“1” and ISO bit =“1”, a packet control internal signal is not output to the transmit control circuit inside the USB function control unit even if SET_IN_BUF_RDY bit of the corresponding endpoint is set to “1” (even if a data packet whose size is equal to the maximum packet size is written to the IN FIFO when AUTO_SET function is enabled). Setting “1” to SET_IN_BUF_RDY bit, which is the packet control signal, is delayed until the next SOF is received so that transmission of the IN FIFO data packet is delayed. The USB function control unit, on detecting that there are transmit packet data at the time of status updating of the IN FIFO, operates artificially as having no transmit packet data until the next SOF packet is detected. On detecting SOF packet, one packet of data which has been set to the IN FIFO is transmitted to the IN token from the host CPU.

•Artificial SOF enable bit

This bit enables the artificial SOF function.

With this bit being set to “1”, when a SOF packet from the host PC has been destroyed due to any cause and no valid SOF packet has been received even after 1ms from the preceding start of the frame, the artificial SOF receive is operated. (And the USB SOF interrupt request, also, occurs.) Therefore, even if the SOF packet is destroyed due to any cause, a new frame can be formed by this function without waiting for the next SOF packet. The artificial SOF receive function is operated once after the valid SOF packet is received twice.

•Artificial SOF status flag

This flag is the artificial SOF function status flag.

This flag is valid when the artificial SOF function has been enabled (Artificial SOF enable bit is “1”).

With this flag being set to “1”, an artificial SOF receive has occurred by the artificial SOF function.

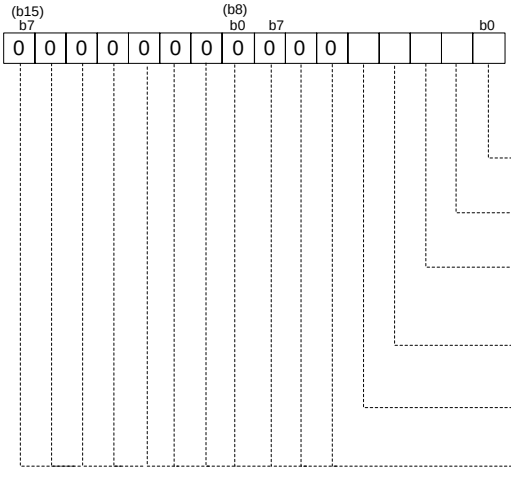
This flag is cleared by setting “1” to CLR_ART_SOF bit .

•Artificial SOF status clear bit

Artificial SOF status flag is cleared to “0” by setting “1” to this bit.

The configuration of USB ISO control register is shown in Figure 2.8.45.

USB ISO Control register



Symbol: USBISOC | Address: 028C16 | When reset: 000016

Bit Symbol	Bit Name	Function	R	W
AUTO_FL	Auto flush bit	0 : Hardware auto flush disabled 1 : Hardware auto flush enabled	O	O
ISO_UPD	ISO Update bit	0 : ISO update disabled 1 : ISO update enabled	O	O
ART_SOF_ENA	Artificial SOF enable bit	0 : Artificial SOF disabled 1 : Artificial SOF enabled	O	O
ART_SOF_SET	Artificial SOF set flag	0 : Not generated by device (Note 1) 1 : Generated by the device	O	X
CLR_ART_SOF	Clear artificial SOF set flag	0 : No action 1 : Clear ART_SOF_SET flag	O (Note 2)	O
Reserved		Must always be “0”	O	O

Note 1: Read only.
Note 2: Always read “0”.

Figure 2.8.45. USB ISO control register

● **USB endpoint x(x=1 to 4) IN control and status register**

•IN_BUF_STS1, IN_BUF_STS0 flags

These flags indicate IN FIFO state.

At the time of writing the data to be transmitted to the host PC in IN FIFO, read these flags to confirm the IN FIFO state.

They are respectively set to “11_2” at the time of resetting. When the corresponding endpoint becomes enable state from disable state, the IN_BUF_STS1 and the IN_BUF_STS0 flags are respectively cleared to “00_2”, automatically. When they are “00_2”, there are no data in IN FIFO. When they are respectively set to “01_2”, there are only one buffer data in double buffer (Invalid for single buffer) . When they are respectively set to “11_2”, there are no space in IN FIFO. (There are one buffer data in single buffer while there are two buffer data in double buffer.) When they are respectively set to “10_2”, it is invalid.

These flags are updated when one of the following events occurs:

- One buffer data of IN FIFO is successfully transmitted to the host PC.
- One buffer data is successfully prepared in IN FIFO

  SET_IN_BUF_RDY bit is set to “1” when writing of one transmit data to the IN FIFO completes. Or, at the time of AUTO_SET enable, when the data count equal to the EPxIMP (or the BUF_SIZ in continuous transfer enable) has been written in the FIFO. However, when a short packet has been written at the time of AUTO_SET enable, these flags are not automatically updated. In such cases, set SET_IN_BUF_RDY bit to “1” by software.
- One buffer data is flushed.

•UNDER_RUN flag

This flag indicates occurrence of an underrun in isochronous transfer. The bit is valid only in isochronous IN transfer. When there are no data packet in IN FIFO at start of the IN token from the host CPU, occurrence of an underrun is recognized and this flag is set to “1”. This flag is cleared to “0” by setting “1” to CLR_UNDER_RUN bit .

•SET_IN_BUF_RDY bit

This bit controls IN FIFO. When there is a space in IN FIFO (when IN_BUF_STS1, IN_BUF_STS0=“00_2” in single buffer, or IN_BUF_STS1, IN_BUF_STS0=“00_2” or “01_2” in double buffer enable), data can be written in the IN FIFO. One transmit data is prepared by setting this bit to “1” after writing the transmit data to the IN FIFO. (To transmit an empty packet, set this bit to “1” without writing any data.) When this bit is set to “1”, the completion of one transmit data ready is notified to the USB function control unit and, simultaneously, the IN FIFO status (IN_BUF_STS1, IN_BUF_STS0 flags) is updated. In the AUTO_SET enable, when a short packet (data whose size is smaller than the EPxIMP value in continuous transfer disable or the BUF_SIZ value in continuous transfer enable) has been written, the IN_BUF_STS1 and IN_BUF_STS0 flags are not automatically updated. In this case, set this bit to “1”.

•CLR_UNDER_RUN bit

The UNDER_RUN flag is cleared to “0” by setting “1” to this bit.

•TOGGLE_INIT bit

This bit initializes data toggle bit required in bulk and interrupt transfer.

When initialization of the data toggle sequence is requested from the host CPU at the time of configuration, etc., set this bit to “1” before starting the IN endpoint communication and initialize PID to DATA0. At this time, the internal read/write pointer of IN FIFO is also initialized.

•FLUSH bit

This bit controls the IN FIFO packet.

Read the IN_BUF_STS1 and IN_BUF_STS0 flags and confirm that there are data in the IN FIFO, and then, set this bit to “1”. When the IN FIFO is flushed, the IN_BUF_STS1 and IN_BUF_STS0 flags are updated as follows:

- When there is one buffer data in IN FIFO, the IN FIFO becomes empty.
  At this time, the IN_BUF_STS1 and IN_BUF_STS0 flags are updated to “00_2”.
- When two buffer data exist in IN FIFO, the older data is flushed.
  At this time, the IN_BUF_STS1 and IN_BUF_STS0 flags are updated to “01_2”. (This indicates that one more buffer data is left inside the IN FIFO.)

The transmit data may be destroyed if this bit is set to “1” during USB transfer.

On completing one buffer data flush, this bit is automatically cleared to “0”.

•INTPT bit

This bit controls transfer mode in interrupt transfer. Only when using the IN endpoint for the rate feedback interrupt transfer, set this bit to “1”.

With this bit being set to “1”, when an IN token is received from the host CPU, IN FIFO data are transmitted regardless of the IN_BUF_STS1 and IN_BUF_STS0 flag states or the data toggle.

Fix this bit at “0” for isochronous transfer, bulk transfer, and normal interrupt transfer.

•ISO bit

This bit controls isochronous transfer. With this bit being set to “1”, the IN endpoint is used for isochronous transfer. Fix this bit at “0” for bulk transfer and interrupt transfer.

•SEND_STALL bit

This bit controls the STALL response to the host CPU.

Set this bit to “1” when the IN endpoint is in STALL state. While this bit is set to “1”, the USB function control unit transmits the STALL handshake concerning all the IN transactions to the host CPU. When the IN endpoint has returned from STALL state, write “0” to clear this bit. The IN endpoint communication is resumed.

-AUTO_SET bit

This bit controls setting of SET_IN_BUF_RDY bit.

With this bit being set to “1”, when one data packet whose is equal to the maximum packet size (EPxIMP set value) has been written to IN FIFO in continuous transmit disable, or, when data equal to the buffer size (byte count set in the BUF_SIZ of the EPxIFC) have been written to IN FIFO in continuous transmit enable, the IN_BUF_STS1 and IN_BUF_STS0 flags are updated without SET_IN_BUF_RDY bit being set to “1”.

However, when a short packet (data whose size is smaller than the EPxIMP value in continuous transfer disable or the BUF_SIZ value in continuous transfer enable) has been written, IN_BUF_STS1 and IN_BUF_STS0 flags are not automatically updated. In such cases, set SET_IN_BUF_RDY bit to “1” by software.

With this bit being set to “0”, set SET_IN_BUF_RDY bit to “1” by software after the transmit data are written to IN FIFO.

The configuration of USB endpoint x (x=1 to 4) IN control and status register is shown in Figure 2.8.46.

USB Endpoint x IN Control and Status register

Symbol: EPxICS (x = 1 - 4)
Address: 029E16, 02A416, 02AA16, 02B016
When reset: 000316

Bit Symbol	Bit Name	Function	R	W
INxCSR0	IN_BUF_STS0 flag	These two bits indicate the EPx IN buffer status Bit1 Bit0	O	X
INxCSR1	IN_BUF_STS1 flag	0 0 : No data set in the IN buffer 0 1 : Single buffer mode: N/A Double buffer mode: one data set in the IN buffer 1 0 : Single buffer mode: N/A Double buffer mode: N/A 1 1 : Single buffer mode: one data set in the IN buffer Double buffer mode: two data sets in the IN buffer	O	X
INxCSR2	UNDER-RUN flag	0 : No underrun detected 1 : Underrun detected	O	x
INxCSR3	SET_IN_BUF_RDY	0 : No action 1 : Data set loaded to the IN buffer (updates IN buffer status flags)	O Note	O
INxCSR4	CLR_UNDER_RUN	0 : No action 1 : Clears UNDER_RUN flag	O Note	O
INxCSR5	TOGGLE_INT	0 : No action 1 : Initialize the next data PID as a DATA0 for transmission	O Note	O
INxCSR6	FLUSH	0 : No action 1 : Flush out one data set	O Note	O
INxCSR7	INTPT	0 : Select non-rate feedback interrupt transfer 1 : Select rate feedback interrupt transfer	O	O
INxCSR8	ISO	0 : Select non-isochronous endpoint 1 : Select isochronous endpoint	O	O
INxCSR9	SEND_STALL	0 : No STALL by CPU 1 : STALL by CPU	O	O
INxCSR10	AUTO_SET	0 : AUTO_SET disabled 1 : AUTO_SET enabled	O	O
Reserved		Must always be set to “0”	O	O

Note: Always read a “0”.

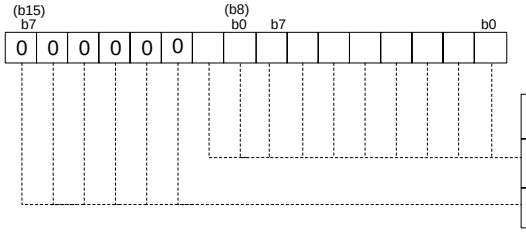
Figure 2.8.46. USB endpoint x(x=1 to 4) IN control and status register

● **USB endpoint x(x=1 to 4) IN MAXP register**

This register indicates endpoint x(x=1 to 4) IN maximum packet size. The default value is 0 byte. When the endpoint is initialized due to any reason such as that the request for setting the endpoint (SET_DESCRIPTOR, SET_CONFIGURATION, SET_INTERFACE, etc.) is received from the host CPU, change the endpoint x IN maximum packet size value by writing in this register. Set a packet size value specified for every transfer type to be used.

The configuration of USB endpoint x(x=1 to 4) IN MAXP register is shown in Figure 2.8.47.

USB Endpoint x IN MAXP register



Symbol: EPxIMP (x = 1 - 4)
Address: 02A0$_{16}$, 02A6$_{16}$, 02AC$_{16}$, 02B2$_{16}$
When reset: 0000$_{16}$

Bit Symbol	Bit Name	Function	R	W
IMAXP9-0	Endpoint x IN maximum packet size	Set the endpoint x IN maximum packet size	O	O
Reserved		Must always be set to “0”	O	O

Figure 2.8.47. USB endpoint x(x=1 to 4) IN MAXP register

● **USB endpoint x(x=1 to 4) IN FIFO configuration register**

This register sets endpoint x(x=1 to 4) IN FIFO.

•BUF_NUM

This bit sets the starting location of the endpoint x(x= 1 to 4) IN FIFO per 64 bytes. For example, when IN FIFO is allocated, starting at the 320th byte, the set value is "000101_2".

•BUF_SIZ

This bit sets one buffer size of the endpoint x(x= 1 to 4) IN FIFO per 64 bytes. For example, when 256 bytes is set, the set value is "0100_2".

•DBL_BUF

With this bit being set to "1", IN FIFO of the corresponding endpoint is changed into double buffer mode. The byte count for a valid IN FIFO becomes twice as much as the value specified by the BUF_SIZ at the time of double buffer. Set carefully not to overlap with the FIFO start position of other endpoints.

•CONTINUE

Set this bit to "1" when continuous transmit is enabled. The bit is valid only in bulk transfer.

The USB function control unit, by dividing one buffer data equal to the byte count set in the BUF_SIZ in the IN FIFO into one packet size (the maximum packet size set in the EPxIMP) units, transmits them one by one to the host PC. (When the last one packet is smaller than the size set in the EPxIMP, it is transmitted as a short packet.) When continuous transmit mode is enabled, the value set in the BUF_SIZ has to be equal to an integral multiple of the maximum packet size.

Pay attention to the following when setting this register:

- Not exceed 3072 bytes in IN FIFO starting location + IN FIFO size.
- Not overlap endpoint FIFOs each other.

The configuration of USB endpoint x(x=1 to 4) IN FIFO configuration register is shown in Figure 2.8.48.

USB Endpoint x IN FIFO register

(b15) b7 (b8) b0 b7 b0

0	0	0	0												

Symbol: EPxIFC (x = 1 - 4)

Address: $02A2_{16}$, $02A8_{16}$, $02AE_{16}$, $02B4_{16}$

When reset: 0000_{16}

Bit Symbol	Bit Name	Function	R	W
BUF_NUM	FIFO buffer start number	Select the starting number for the EPx IN FIFO (in units of 64 bytes) 000000 : buffer stating location = 0 000001 : buffer stating location = 64 000010 : buffer stating location = 128 ..... 101111 : buffer stating location = 3008 (last starting number)	O	O
BUF_SIZ	FIFO buffer size	Select the buffer size for the EPx IN FIFO (in units of 64 bytes) 0000 : buffer stating location = 64 0001 : buffer stating location = 128 0010 : buffer stating location = 192 ..... 1111 : buffer stating location = 1024 (largest buffer size)	O	O
DBL_BUF	Double buffer mode	0 : Disabled 1 : Enabled	O	O
CONTINUE	Continuous transfer mode	0 : Disabled 1 : Enabled (Note)	O	O
Reserved		Must always be set to "0"	O	O

Note: Valid for bulk transfer type only.

Figure 2.8.48. USB endpoint x(x=1 to 4) IN FIFO configuration register

(2) Bulk Transfer: Endpoints 1 to 4 Transmit

● **Setting of Transfer Type**

When endpoints 1 to 4 IN are used for bulk transfer, both ISO bit and INTPT bit of USB endpoint x IN control and status register are set to “0” for bulk transfer setting.

Also, for initialization of toggle sequence bit in bulk transfer, set TOGGLE_INIT bit to “1” and initialize PID to DATA0.

In order to enable double buffer mode and continuous transmit mode, use USB endpoint x IN FIFO configuration register for setting.

In order to use the AUTO_SET function, set AUTO_SET bit of USB endpoint x IN control and status register to “1”.

● **Transmit Data Preparation**

The transmit data has to be beforehand prepared in IN FIFO in order to transmit data. Prepare one packet data (Note 1) to the IN FIFO in the following procedure:

1: Confirm that there is a packet space in the IN FIFO.

Read the IN_BUF_STS1 and IN_BUF_STS0 flags and confirm that they are either “00_2” (the IN FIFO empty) or “01_2” (writable of second data in double buffer).

2: Write one packet data (Note 1) to be transmitted to the IN FIFO.

Every time 1-byte data are written to the IN FIFO, the internal write pointer is automatically incremented by one. Contents of the internal write pointer cannot be read. For transmitting an empty packet (with 0 data length), do not write data to the IN FIFO.

3: Set SET_IN_BUF_RDY bit to “1”.

At this time, the IN FIFO status is updated as follows (the IN_BUF_STS1 and IN_BUF_STS0 flags are updated) and the transmit preparation is completed:

- In Single Buffer Mode

  The IN_BUF_STS1 and IN_BUF_STS0 flags are updated from “00_2” (the IN FIFO empty) to “11_2” (the IN FIFO full).

- In Double Buffer Mode

  When the first packet data (Note 1) of double buffer is written while there is a space in IN FIFO, the IN_BUF_STS0 and IN_BUF_STS1 flags are updated from “00_2” to “01_2”, indicating that the second packet data (Note 1) is ready to be written to the IN FIFO. In this case, second packet data can be continuously prepared.

  When there is only the first packet data (Note 1) in the IN FIFO and second packet data (Note 1) is written, the IN_BUF_STS0 and IN_BUF_STS1 flags are updated from “01_2” to “11_2”, indicating that no more data can be written to the IN FIFO.

The USB function control unit transmits one packet data (Note 1) in the next IN token.

Note 1: In continuous transfer enable, read the description by substituting the underlined part with “buffer data”. As for one buffer data, when SET_IN_BUF_RDY bit is set to “1” after data less than the value set in the BUF_SIZ are written to IN FIFO, one buffer data transmit is prepared. Also, the BUF_SIZ has to be equal to an integral multiple of the EPxIMP.

While the AUTO_SET is enabled (AUTO_SET bit is “1”), when one data packet whose is equal to the maximum packet size (EPxIMP set value) has been written to IN FIFO in continuous transmit disable, or, when data equal to the buffer size (byte count set in the BUF_SIZ of the EPxIFC) has been written to IN FIFO in continuous transmit enable, the IN_BUF_STS1 and IN_BUF_STS0 flags are updated without SET_IN_BUF_RDY bit being set to “1”. However, when a short packet (data whose size is smaller than the EPxIMP value in continuous transfer disable or the BUF_SIZ value in continuous transfer enable) has been written, the IN_BUF_STS1 and IN_BUF_STS0 flags are not automatically updated. In such cases, set the SET_IN_BUF_RDY bit to “1” by software.

● **Transmit Operation**

On completing transmitting of one packet data (Note 2) to the host, the IN_BUF_STS0 and IN_BUF_STS1 flags are automatically updated. In single buffer mode (when double buffer mode bit is “0”), these flags are updated from “11_2” to “00_2”. In double buffer mode, they are updated as follows:

- While there are two packet data (Note 2) in IN FIFO, the IN_BUF_STS0 and IN_BUF_STS1 flags are updated from “11_2” to “01_2” when one of the data is transmitted, indicating that one more transmit data is left inside the IN FIFO.
- When there is one packet data (Note 2) in IN FIFO, the IN_BUF_STS0 and IN_BUF_STS0 flags are updated from “01_2” to “00_2” when the data are transmitted, indicating that the IN FIFO becomes empty.

Note 2: In continuous transfer enable, read the description by substituting the underlined part with “buffer data”. The USB function control unit transmits the transmit data in sequence by one packet size (the maximum packet size set in the EPxIMP). (When the last one packet is smaller than the size set in the EPxIMP, it is received as a short packet.)

When IN token is received from the host CPU while SEND_STALL bit is set to “1”, STALL response is automatically returned.

When IN token is received from the host CPU while there are no packet data in the IN FIFO, NAK response automatically is returned.

When IN token is received from the host CPU while there are packet data in the IN FIFO, data are transmitted by using the current data toggle sequence bit. On completing one packet data transmit (on receiving ACK from the host CPU), the IN FIFO status is updated (the IN_BUF_STS1 and IN_BUF_STS0 flags are updated) and data toggle sequence bit is toggled (DATA0 → DATA1, or DATA1 → DATA0). At this time, the endpoint x IN interrupt request occurs. When one packet data has been unsuccessfully transmitted (ACK not received from the host CPU), the data are re-transmitted in the next IN token (the same data are transmitted in the same toggle).

(3) Isochronous Transfer: Endpoints 1 to 4 Transmit

● **Type of Transmit Transfer**

When endpoints 1 to 4 IN are used for isochronous transfer, ISO bit and INTPT bit of USB endpoint x IN control and status register are respectively set to “1” and to “0”, for isochronous transfer setting.

● **Transmit Data Preparation**

The endpoint x IN packet data preparation procedure in the isochronous transfer is same as the bulk transfer.

Refer to “● Transmit Data Preparation” of “(2) Bulk Transfer: Endpoints 1 to 4 Transmit” (Continuous transfer is valid for the bulk transfer only).

● **Transmit Operation**

A packet whose PID is DATA0 is transmitted to the IN token from the host CPU.

When IN token is received from the host CPU while there are no data in IN FIFO, an empty packet (with 0 data length) is automatically transmitted, an underrun error occurs, and the UNDER_RUN flag is set to “1”.

When the UNDER_RUN flag is set to “1” while the error interrupt is enabled by USB function interrupt enable register, an error interrupt request occurs (INTST8 is set to “1”).

When IN token is received from the host CPU while there are packet data in IN FIFO, data are transmitted. At this time, the IN FIFO status is updated and the endpoint x IN interrupt request occurs. In isochronous transfer, timing in which data are transmitted to the IN token from the host CPU differs by setting of ISO_UPDATE bit. When ISO_UPDATE bit is set to “1”, setting “1” to SET_IN_BUF_RDY bit, which is the packet control signal, is delayed until the next SOF packet is received so that transmission of the IN FIFO data packet is delayed. At this time, whether there are actual packet data or not, it is replied to the IN token that there are no packet data in the IN FIFO until the next SOF packet is detected.

Also, for flushing the IN FIFO in isochronous IN transfer by using software, the AUTO FLUSH function is used.

While ISO_UPDATE bit =“1”, AUTO_FLUSH bit =“1”, and ISO bit (INxCSR8)=“1”, the USB function control unit, at the time of detecting a SOF packet (from the host PC or on the artificial SOF), automatically flushes old data packet inside the IN FIFO if both the IN_BUF_STS1 and the IN_BUF_STS0 flags are “1” (IN FIFO full state). In isochronous transfer for double buffer, use the AUTO FLUSH function.

(4) Interrupt Transfer: Endpoints 1 to 4 Transmit

● **Setting of Transfer Type**

Interrupt transfer setting has two kinds including the normal interrupt transfer and the rate feedback interrupt transfer.

When endpoints 1 to 4 IN are used for the normal interrupt transfer, ISO bit and INTPT bit of USB endpoint x IN control and status register are respectively to “0”.

When the isochronous device has the rate feedback function and endpoints 1 to 4 IN are used for rate feedback interrupt transfer, not only ISO bit and INTPT bit of USB endpoint x IN control and status register are respectively set to “0” and to “1” but also double buffer mode enable bit of USB endpoint x IN FIFO configuration register is set to “0” so that single buffer is enabled.

Also, for initialization of toggle sequence bit in interrupt transfer, set TOGGLE_INIT bit to “1” and initialize PID to DATA0.

● **Transmit Data Preparation**

Normal Interrupt Transfer:

The endpoint x IN packet data preparation procedure in the normal interrupt transfer is same as the bulk transfer.

Refer to “● Transmit Data Preparation“ of “(2) Bulk Transfer: Endpoints 1 to 4 Transmit” (Continuous transfer is valid for the bulk transfer only).

Rate Feedback Interrupt Transfer:

In real application, transmit data to the host CPU has to be always prepared. Prepare one transmit data to the IN FIFO in the following procedure:

For details of the following 1 and 2, refer to the single buffer mode parts in “● Transmit Data Preparation (2, 3)” of “(2) Bulk Transfer: Endpoints 1 to 4 Transmit” (Continuous transfer is valid for the bulk transfer only).

1: Write one packet data to be transmitted to the IN FIFO. At the time of writing the data, pay attention to the timing so that an IN token is not received from the host. Every time 1-byte data are written to the IN FIFO, the internal write pointer is automatically incremented by one. Contents of the internal write pointer cannot be read. For transmitting an empty packet (with 0 data length), do not write data to the IN FIFO.

2: Set SET_IN_BUF_RDY bit to “1” after writing of the data to the IN FIFO are completed.

At this time, the IN FIFO state is updated and one packet transmit is prepared. The USB function control unit transmits this data to the IN token until next transmit data are updated.

● **Transmit Operation**

Normal Interrupt Transfer:

The endpoint x IN transmit operation in the normal interrupt transfer is same as the bulk transfer.

Refer to “● Transmit Operation” of “(2) Bulk Transfer: Endpoints 1 to 4 Transmit”.

Rate Feedback Interrupt Transfer:

In real application, rate feedback interrupt transfer always has data to be transmitted to the host. Therefore, the device does not repond with NAK to the IN token from the host in this transfer. On receiving IN token from the host CPU, the IN FIFO data are always transmitted in the current data sequence bit regardless of the IN_BUF_STS0 and IN_BUF_STS1 values. Except this point, the transmit operation is the same as the normal interrupt transfer.

When IN token is received from the host CPU while SEND_STALL bit being set to “1”, STALL response is automatically returned. On receiving IN token from the host CPU, the IN FIFO data are transmitted in the current data sequence bit . On completing one data transmit (on receiving ACK from the host CPU), the IN FIFO status is updated, data toggle sequence bit is toggled (DATA0 →DATA1 or DATA1→DATA0), and the endpoint x IN interrupt request occurs. At this time, unlike the normal interrupt transfer, the IN FIFO data are not deleted, which is retained until the next packet data are updated. When one data transmit has not been unsuccessfully completed (an ACK not received from the host CPU), the data are re-transmitted in the next IN token (the same data are transmitted in the same toggle).

(5) Precautions for Transmit

● Writing to IN FIFO

Be sure to confirm that there is a space in the IN FIFO before writing data to the IN FIFO in preparation for packet data to the IN FIFO.

The IN FIFO state is indicated by the IN_BUF_STS1 and the IN_BUF_STS0 flags. Based on these flags states, determine the count of data packets set in the IN FIFO.

The IN FIFO status (IN_BUF_STS1 and IN_BUF_STS0 flags) is updated when transmit data are prepared in the IN FIFO (SET_IN_BUF_RDY bit is set to “1”), when transmitting of one data to the host CPU is completed, or when data inside the IN FIFO have been flushed (AUTO_FLUSH bit or FLUSH bit has functioned.)

Table 2.8.4. Status on Endpoint 1 to 4 IN FIFOs

IN_BUF_STS1	IN_BUF_STS0	Single buffer [Specify IN FIFO size by the BUF_SIZ*1]	Double buffer [IN FIFO size = (The number of bytes specified by the BUF_SIZ*1) × 2]
0	0	No data Space equal to one buffer	No data Space equal to two buffer
0	1	Invalid	Invalid
1	0	Invalid	One data set in the IN FIFO Space equal to one buffer
1	1	One data set in the IN FIFO No space in the IN FIFO	Two data set in the IN FIFO No space in the IN FIFO

*1: Bits 6 to 9 of EPxIFC.

● PID Initialization

When TOGGLE_INIT bit is set to “1”, the read/write counter inside the FIFO is initialized. To initialize the PID, set TOGGLE_INIT bit to “1” in the IN FIFO is empty state (the IN_BUF_STS0 and IN_BUF_STS1 flags are “00_2”).

(6) USB Transmit (Endpoints 1 to 4 IN): Example

The endpoints 1 to 4 IN transmit packet prepare routine (continuous transfer disable) is shown in Figure 2.8.49. In addition to packet prepare process, error process by the UNDER_RUN flag is required in isochronous transfer.

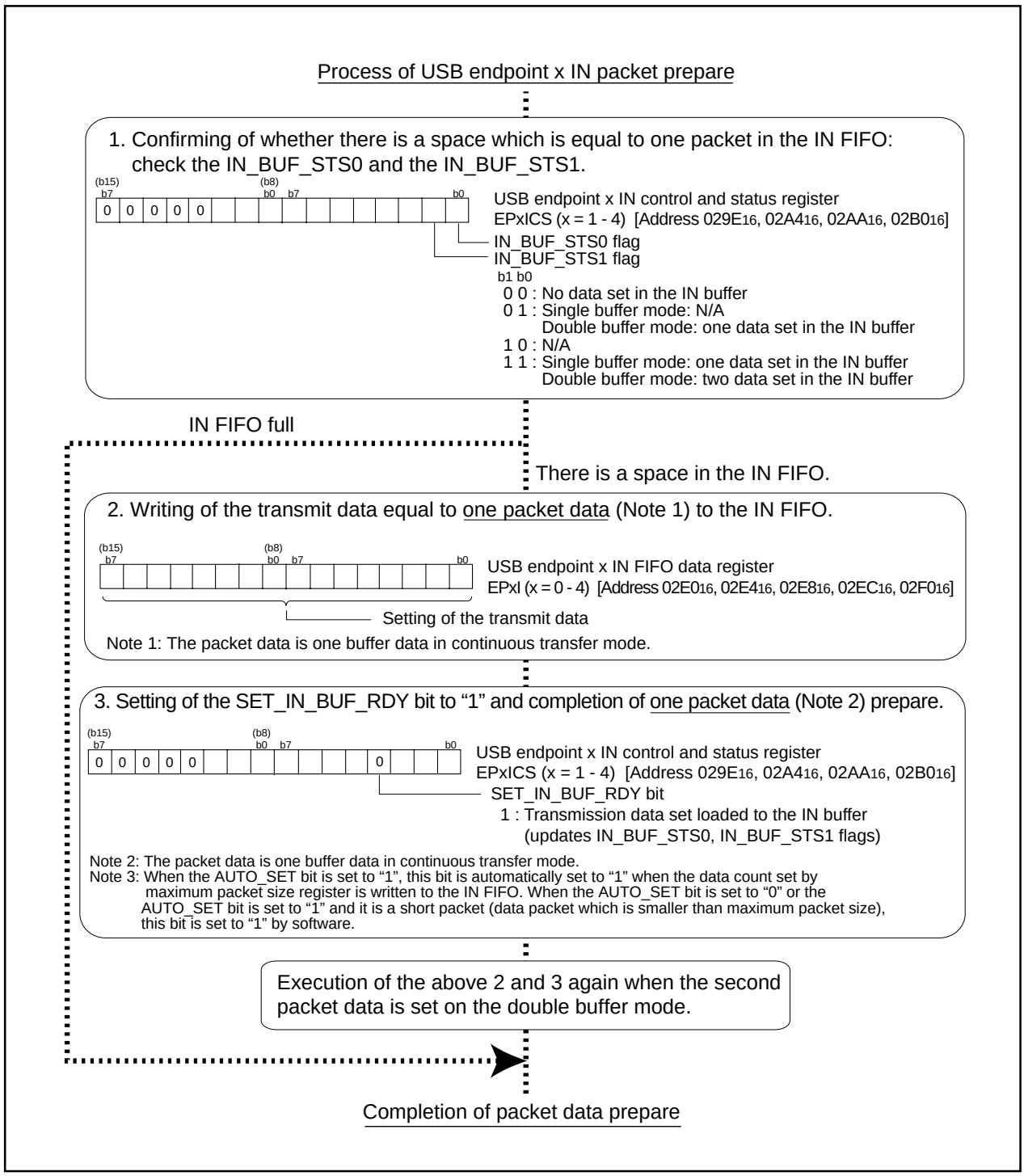

Process of USB endpoint x IN packet prepare
1. Confirming of whether there is a space which is equal to one packet in the IN FIFO: check the IN_BUF_STS0 and the IN_BUF_STS1.
(b15) b7 (b8) b0 b7 b0
0 0 0 0 0
USB endpoint x IN control and status register
EPxICS (x = 1 - 4) [Address 029E16, 02A416, 02AA16, 02B016]
IN_BUF_STS0 flag
IN_BUF_STS1 flag
b1 b0
0 0 : No data set in the IN buffer
0 1 : Single buffer mode: N/A
Double buffer mode: one data set in the IN buffer
1 0 : N/A
1 1 : Single buffer mode: one data set in the IN buffer
Double buffer mode: two data set in the IN buffer
IN FIFO full
There is a space in the IN FIFO.
2. Writing of the transmit data equal to one packet data (Note 1) to the IN FIFO.
(b15) b7 (b8) b0 b7 b0
USB endpoint x IN FIFO data register
EPxI (x = 0 - 4) [Address 02E016, 02E416, 02E816, 02EC16, 02F016]
Setting of the transmit data
Note 1: The packet data is one buffer data in continuous transfer mode.
3. Setting of the SET_IN_BUF_RDY bit to “1” and completion of one packet data (Note 2) prepare.
(b15) b7 (b8) b0 b7 b0
0 0 0 0 0 0
USB endpoint x IN control and status register
EPxICS (x = 1 - 4) [Address 029E16, 02A416, 02AA16, 02B016]
SET_IN_BUF_RDY bit
1 : Transmission data set loaded to the IN buffer
(updates IN_BUF_STS0, IN_BUF_STS1 flags)
Note 2: The packet data is one buffer data in continuous transfer mode.
Note 3: When the AUTO_SET bit is set to “1”, this bit is automatically set to “1” when the data count set by maximum packet size register is written to the IN FIFO. When the AUTO_SET bit is set to “0” or the AUTO_SET bit is set to “1” and it is a short packet (data packet which is smaller than maximum packet size), this bit is set to “1” by software.
Execution of the above 2 and 3 again when the second packet data is set on the double buffer mode.
Completion of packet data prepare

Figure 2.8.49. Endpoint 1 to 4 IN packet prepare routine

2.8.8 USB Operation (Interface with DMAC Transfer)

The M30245 group can select a USB (USB0/USB1/USB2/USB3) as the DMA request factor. The USB0 corresponds to DMA0, USB1 to DMA1, USB2 to DMA2, and USB3 to DMA3. The DMA request factor origin of USB0/USB1/USB2/USB3 is also set by setting any one of endpoints 1 to 4 IN/OUT factors to USB DMAx(x=0 to 3) request register .

The DMA request factor of USB0/USB1/USB2/USB3 occurs, under particular conditions, not only on occurrence of an interrupt request of each endpoint but also on write/read to/from IN/OUT FIFO.

(1) Related Registers

● **USB DMAx(x=0 to 3) request register**

This register sets the DMA request factor origin of USB0/USB1/USB2/USB3. When, under particular conditions, write/read to/from the FIFO of the endpoint selected by this register or an event such as the endpoint's interrupt request occurs, a DMA request occurs. This register can be set “1” only to 1 bit. When multiple bits are simultaneously set to “1”, the setting becomes invalid. Other DMA related registers also need to be set before a valid value is set; for example, “00011_2” (USB0/USB1/USB2/USB3) is set to DMA request cause select bits (b4,b3,b2,b1,b0) of DMAx(x=0 to 3) request cause select register (addresses $03B8_{16}$, $03BA_{16}$, $03B0_{16}$, $03B2_{16}$).

The configuration of USB DMAx(x=0 to 3) request register is shown in Figure 2.8.50.

USB DMAx Request registers

(b15) b7 (b8) b0 b7 b0

0	0	0	0	0	0					0					0

Symbol: USBDMAx (x=0 to 3)

Address: 0290_{16}, 0292_{16}, 0294_{16}, 0296_{16}

When reset: 0000_{16}

Bit Symbol	Bit Name	Function	R	W
Reserved		Must always be set to “0”	O	O
DMAxR1	EP1 IN FIFO write request select bit	1 : Selected 0 : Not selected	O	O
DMAxR2	EP2 IN FIFO write request select bit			
DMAxR3	EP3 IN FIFO write request select bit			
DMAxR4	EP4 IN FIFO write request select bit			
Reserved		Must always be set to “0”	O	O
DMAxR6	EP1 OUT FIFO read request select bit	1 : Selected 0 : Not selected	O	O
DMAxR7	EP2 OUT FIFO read request select bit			
DMAxR8	EP3 OUT FIFO read request select bit			
DMAxR9	EP4 OUT FIFO read request select bit			
Reserved		Must always be set to “0”	O	O

Figure 2.8.50. USB DMAx(x=0 to 3) request register

(2) DMA Request by Endpoint x OUT

● **DMA Request Factors**

When endpoint 1 to 4 OUT FIFO write request select bit is set to the DMA request factor origin of USB0/USB1/USB2/USB3, the DMA request factor includes the following three kinds. On occurrence of an event when all the specified conditions have been satisfied for each factor, the DMA request of DMA0/DMA1/DMA2/DMA3 occurs.

•Factor 1

Conditions:

- DMA enable bit of DMAi control register is set to “1” (enable).
- Any one of endpoint x(x=1 to 4) OUT FIFO read request select bit in USB DMAx(x=0 to 3) request register is set to “1”. The other bits are set to “0” (valid setting).

Event:

The OUT FIFO state of the endpoint x OUT which is set in USB DMAx(x=0 to 3) request register has been updated, and the OUT_BUF_STS1 and OUT_BUF_STS0 flags are set to “11_2” at the time of single buffer and “10_2” at the time of double buffer. (When data of one or more buffers are received in OUT FIFO. At this time, when one packet receive is completed, the endpoint x OUT interrupt request simultaneously occurs.)

•Factor 2

Conditions:

- DMA enable bit of DMAi control register is set to “1” (enable).
- The OUT_BUF_STS1 and OUT_BUF_STS0 flags of endpoint x OUT which is set in USB DMAx(x=0 to 3) request register are set to “10_2” or “11_2”. (When data of one or more packets are received in OUT FIFO.)
- There is no selection of USB DMAx(x=0 to 3) request register (“00_{16}”).

Event:

Any one of endpoint x(x=1 to 4) OUT FIFO read request select bit in USB DMAx(x=0 to 3) request register is set to “1”. The other bits are set to “0” (valid setting).

•Factor 3

Conditions:

- DMAenable bit of DMAi control register is set to “1” (enable).
- The OUT_BUF_STS1 and OUT_BUF_STS0 flags of endpoint x OUT which is set in USB DMAx(x=0 to 3) request register are set to “10_2” or “11_2”. (When data of one or more packets are received in OUT FIFO.)
- Any one of endpoint x(x=1 to 4) OUT FIFO read request select bit in USB DMAx(x=0 to 3) request register is set to “1”. The other bits are set to “0” (valid setting).

Event:

1-byte (1-word) data is read from the endpoint x OUT FIFO which is set in USB DMAx(x=0 to 3) request register .

● **Reading of Endpoint x OUT FIFO in DMA Transfer**

The DMA request factor of USB0/USB1/USB2/USB3 corresponds to read from the endpoints 1 to 4 OUT FIFO (Factor 3). Therefore, with endpoint x OUT FIFO being specified to the DMA source pointer and the transfer source address direction being fixed, when DMA transfer is executed by Factor 1 (Factor 2 or Factor 3), Factor 3 occurs. Therefore, when one buffer data (one packet data) is read from OUT FIFO by DMA transfer, it is possible that the 1st byte (1st word) data is DMA transferred by Factor 1 (Factor 2 or Factor 3) and the other data, starting from the 2nd byte (2nd word) up to the last byte (last word), are DMA transferred by Factor 3.

For details of DMA transfer, refer to “Chapter 2. DMAC”.

(3) DMA Request by Endpoint x IN

● **DMA Request Factor**

When endpoint x(x=1 to 4) IN FIFO write request select bit is set to the DMA request factor origin of USB0/USB1/USB2/USB3, the DMA request factor includes the following three kinds. On occurrence of an event when all the specified conditions have been satisfied for each factor, the DMA request of DMA0/DMA1/DMA2/DMA3 occurs.

•Factor 1

Conditions:

- DMA enable bit of DMAi control register is set to “1” (enable).
- Any oneendpoint x(x=1 to 4) IN FIFO write request select bit in USB DMAx(x=0 to 3) request register is set to “1”. The other bits are set to “0” (valid setting).

Event:

The IN FIFO state of the endpoint x IN which is set in USB DMAx(x=0 to 3) request register has been updated, and the IN_BUF_STS1 and IN_BUF_STS0 flags are set to “00_2” at the time of single buffer and “01_2” at the time of double buffer. (When there are the space of one or more packets in the IN FIFO. At this time, when one packet transfer is completed, the endpoint x IN interrupt request simultaneously occurs.)

•Factor 2

Conditions:

- DMA enable bit of DMAi control register is set to “1” (enable).
- The IN_BUF_STS1 and IN_BUF_STS0 flags of endpoint x IN which is set in USB DMAx(x=0 to 3) request register are set to “00_2” or “01_2”. (When there are the space of one or more packets in the IN FIFO.)
- There is no selection of USB DMAx(x=0 to 3) request register (“00_{16}”).

Event:

Any one endpoint x(x=1 to 4) IN FIFO write request select bit in USB DMAx(x=0 to 3) request register is set to “1”. The other bits are set to “0” (valid setting).

•Factor 3

Conditions:

- DMA enable bit of DMAi control register is set to “1” (enable).
- The IN_BUF_STS1 and IN_BUF_STS0 flags of endpoint x IN which is set in USB DMAx(x=0 to 3) request register are set to “00_2” or “01_2”. (When there are the space of one or more packets in the IN FIFO.)
- Any one of endpoint x(x=1 to 4) IN FIFO write request select bit in USB DMAx(x=0 to 3) request register is set to “1”. The other bits are set to “0” (valid setting).

Event:

1-byte (1-word) data is written in the endpoint x IN FIFO which is set in USB DMAx(x=0 to 3) request register .

● **DMA Transfer to Endpoint x IN FIFO**

The DMA request factor of USB0/USB1/USB2/USB3 corresponds to write in the endpoints 1~4 IN FIFO (Factor 3). Therefore, with endpoint x IN FIFO being specified to the DMA destination pointer and the transfer destination address direction being fixed, when DMA transfer is executed by Factor 1 (Factor 2 or Factor 3), Factor 3 occurs. Therefore, when one buffer data is written in IN FIFO by DMA transfer, it is possible that the 1st byte (1st word) data is DMA transferred by Factor 1 (Factor 2 or Factor 3) and the other data, starting from the 2nd byte (2nd word) up to the last byte (last word), are DMA transferred by Factor 3.

For details of DMA transfer, refer to “Chapter 2.10 DMAC”.

2.8.9 Precautions for USB

(1) USB Communication

"In applications requiring high-reliability, we recommend providing the system with protective measures such as USB function initialization by software or USB reset by the host to prevent USB communication from being terminated unexpectedly, for example due to external causes such as noise."

(2) Peripheral Circuit

The peripheral circuit block diagram is shown in Figure 2.8.51, the passive part of LPF pin is shown in Figure 2.8.52 and the connection diagram of decoupling capacitor is shown in Figure 2.8.53.

•USB2.0 specification specifies the driver impedance 28~44Ω (See 7.1.1.1 “Full-speed (12Mb/s) Driver Characteristics”). Connect a serial resistor (recommended value: 27~33Ω) to the USB D+ pin and the USB D- pin to satisfy this specification. Also connect, if required, the capacitors between the USB D+ pin/USB D- pin and the Vss pin. These capacitors control ringing or adjust the times of rising/falling and the crossover point of D+/D-. As the numerical values and the configuration of the peripheral components need to be adjusted according to differences in characteristic impedance and layout of the mount printed circuit board. Therefore, fully evaluate on the system in use and observe waveforms before adjusting the connection or disconnection and the values of the resistance and the capacitor.

•When the USB Attach/Detach function is not used, connect the UVcc pin and the USB D+ pin via a 1.5kΩ resistance (D+ line pull-up timing depends on the UVCC pin).
When the USB Attach/Detach function is used, connect the P90/ATTACH pin and the USB D+ pin via a 1.5kΩ resistance. Irrespective of use of the USB Attach/Detach function, connect the UVCC pin to the power supply. In addition, the time required for the host PC to recognize the USB Attach/Detach varies depending on the whole system state such as substrate resistance components, capacitance components, USB cable capacitance, and substrate characteristics and processing speed of the host. Fully evaluate on the system subject to actual use.

•Connect all the passive parts of the LPF pin as close to the LPF pin as possible.

•Connect an insulating connector (ferrite beads) between the AVss pin and the digital Vss, between the AVcc pin and the digital Vcc. Also at this time, when an A/D converter is used, connect the Vref pin to the AVcc pin.

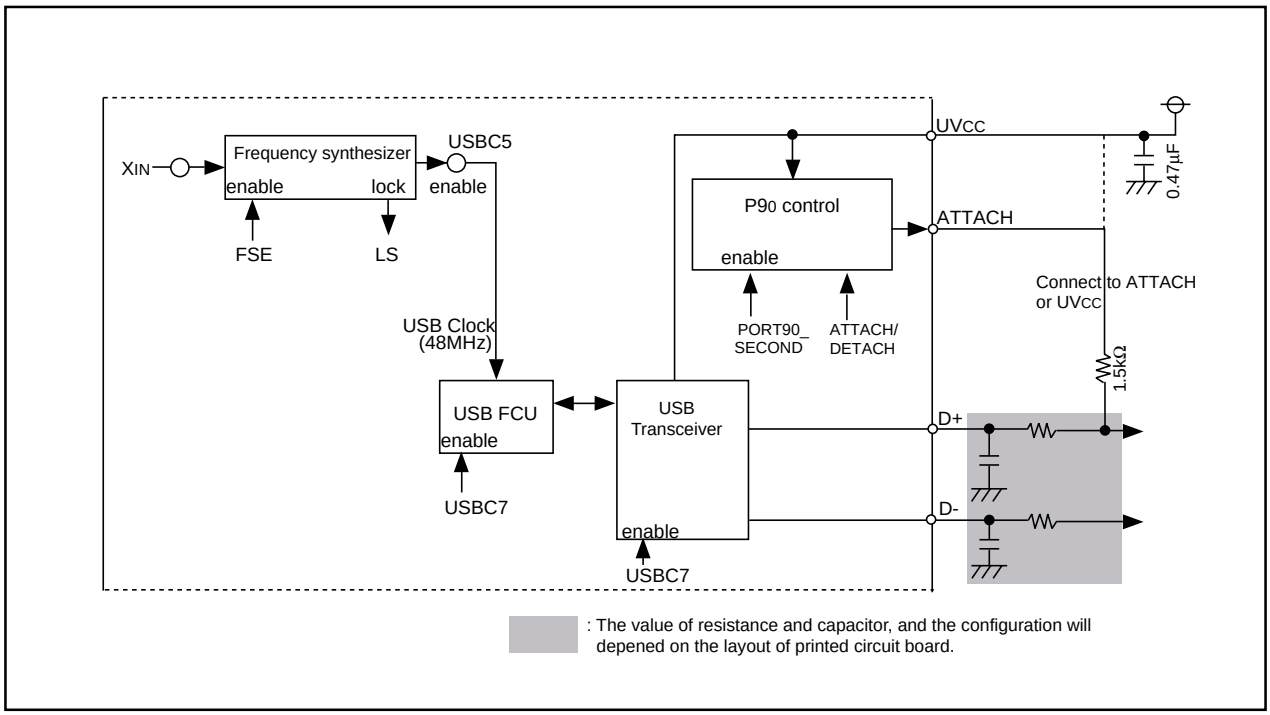


Figure 2.8.51. Peripheral circuit block diagram

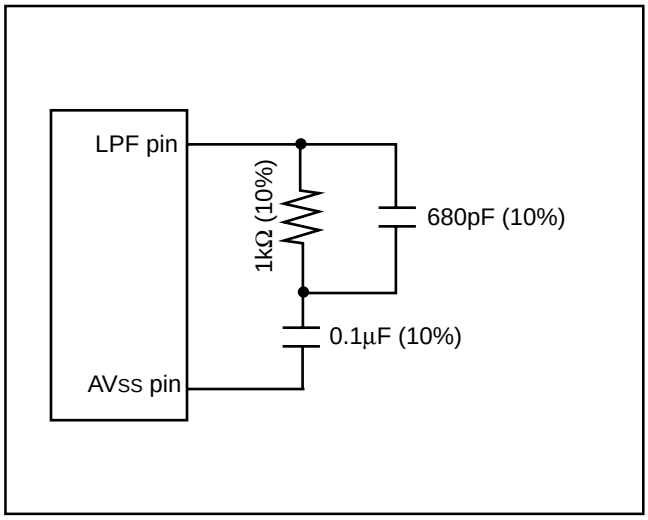


Figure 2.8.52. Passive part of LPF pin

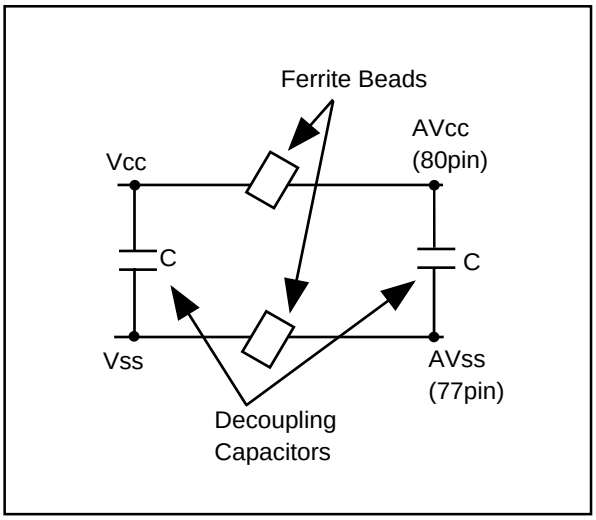


Figure 2.8.53. Decoupling capacitor

(3) Register, Bit

- When the USB reset interrupt request occurs, all the USB internal registers become reset state. To resume communication, each endpoint needs to be initialized.
- All the USB related registers (16-bit registers) except USB endpoint x(x=0 to 4) IN FIFO data register (EPxI), USB endpoint x(x=0 to 4) OUT FIFO data register (EPxO), USB control register (USBC), and USB attach/detach register (USBAD) are available for word access and byte access. The EPxI and the EpxO are only available for word access or byte access to the lower bytes. The USBC and the USBAD of 8-bit registers are only available for byte access. After software reset, contents of all the USB related registers are retained.
- While the USB clock is held disabled in suspend mode, writing in the USB internal registers (other than USBC, USBAD, and frequency synthesizer-related registers) is disabled.

(4) Packet Data Destruction

- When FLUSH bit of endpoint x OUT control and status register (EPxOCS) is set to “1” during USB transfer, the receive data may be destroyed. Be sure to set FLUSH bit of the EPxOCS to “1” only when there are data in OUT FIFO (OUT_BUF_STS1 and OUT_BUF_STS0 are set to “10_2” or “11_2”).
- When FLUSH bit of USB endpoint x IN control and status register (EPxICS) is set to “1” during USB transfer, the transmit data may be destroyed. Be sure to read the IN_BUF_STS1 and the IN_BUF_STS0 flags and to confirm that there are data in IN FIFO before setting FLUSH bit of the EPxICS to “1”.

  In isochronous transfer, use AUTO_FLUSH bit (bit 0 of address $028C_{16}$).

2.9 A/D Converter

2.9.1 Overview

The A/D converter used in the M30245 group operates on a successive conversion basis. The following is an overview of the A/D converter.

(1) Mode

The A/D converter operates in one of five modes:

(a) One-shot mode

Carries out A/D conversion on input level of one specified pin only once.

(b) Repetition mode

Repeatedly carries out A/D conversion on input level of one specified pin.

(c) One-shot sweep mode

Carries out A/D conversion on input level of two or more specified pins only once.

(d) Repeated sweep mode 0

Repeatedly carries out A/D conversion on input level of two or more specified pins.

(e) Repeated sweep mode 1

Repeatedly carries out A/D conversion on input level of two or more specified pins. This mode is different from the repeated sweep mode 0 in that weights can be assigned to specifing pins control the number of conversion times.

(2) Operation clock

The operation clock can be selected from the following: f_{AD}, divide-by-2 f_{AD}, divide-by-3 f_{AD}, and divide-by-4 f_{AD}. The f_{AD} frequency is equal to that of the CPU's main clock ($f_{AD}=f(X_{IN})$). Set the operation clock to 10 MHz or lower. When $f(X_{IN})$ is over 10 MHz, the f_{AD} frequency must be under 10 MHz by dividing.

(3) Conversion time

Number of conversion for A/D convertor varies depending on resolution as given. Table 2.9.1 shows relation between the A/D converter operation clock and conversion time.

Sample & Hold function selected:

33 ϕAD cycles for 10-bit resolution, or 28 ϕAD cycles for 8-bit resolution

No Sample & Hold function:

59 ϕAD cycles for 10-bit resolution, or 49 ϕAD cycles for 8-bit resolution

Table 2.9.1. Conversion time every operation clock

Frequency select bit 1		1	0	0	1
Frequency select bit 0		0	0	1	1
A/D converter's operation clock		$\phi_{AD} = f_{AD}/3$	$\phi_{AD} = f_{AD}/4$	$\phi_{AD} = f_{AD}/2$	$\phi_{AD} = f_{AD}$
Min. conversion cycles (Note 1)	8-bit mode	28 × ϕAD			
	10-bit mode	33 × ϕAD			
Min. conversion time (Note 2)	8-bit mode	8.4µs	11.2µs	5.6µs	2.8µs
	10-bit mode	9.9µs	13.2µs	6.6µs	3.3µs

Note 1: The number of conversion cycles per one analog input pin.

Note 2: The conversion time per one analog input pin (when $f_{AD} = f(X_{IN}) = 10$ MHz)

(4) Functions selection

(a) Sample & Hold function

Sample & Hold function samples input voltage when A/D conversion starts and carries out A/D conversion on the voltage sampled. When A/D conversion starts, input voltage is sampled for 3 cycles of the operation clock. When the Sample & Hold function is selected, set the operation clock for A/D conversion to 1 MHz or higher.

(b) 8-bit A/D to 10-bit A/D switching function

Either 8-bit resolution or 10-bit resolution can be selected. When 8-bit resolution is selected, the 8 higher-order bits of the 10-bit A/D are subjected to A/D conversion. The equations for 10-bit resolution and 8-bit resolution are given below:

10-bit resolution $(Vref \times n / 2^{10}) - (Vref \times 0.5 / 2^{10})$ (n = 1 to 1023), 0 (n = 0)

8-bit resolution $(Vref \times n / 2^{8}) - (Vref \times 0.5 / 2^{10})$ (n = 1 to 255), 0 (n = 0)

(c) A/D conversion by external trigger

The user can select software or an external pin input to start A/D conversion.

(d) Connecting or cutting Vref

Cutting Vref allows decrease of the current flowing into the A/D converter. To decrease the microcomputer's power consumption, cut Vref. To carry out A/D conversion, start A/D conversion 1 µs or longer after connecting Vref.

(5) Input to A/D converter and the relation of direction register

To use the A/D converter, set the direction register of the relevant port to input.

(6) Pins related to A/D converter

(a) AN0 pin through AN7 pin	Input pins of the A/D converter
(b) AVcc pin	Power source pin of the analog section
(c) VREF pin	Input pin of reference voltage
(d) AVss pin	GND pin of the analog section
(e) $\overline{\text{ADTRG}}$ pin	Trigger input pin of the A/D converter

(7) A/D converter related registers

Figure 2.9.1 shows the memory map of A/D converter-related registers, and Figures 2.9.2 and 2.9.3 show A/D converter-related registers.

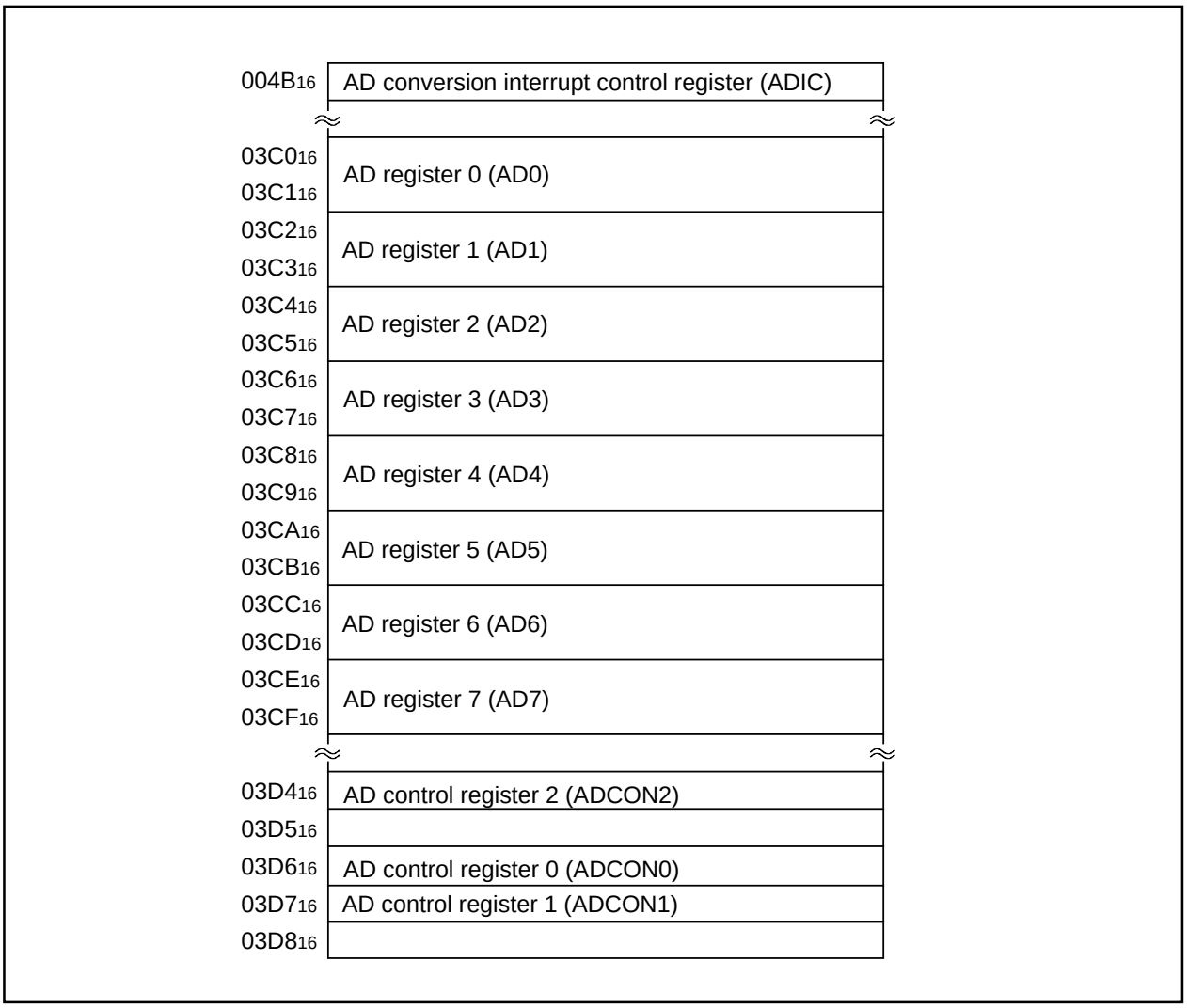


Figure 2.9.1. Memory map of A/D converter-related registers

AD control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: ADCON0 — Address: 03D6$_{16}$ — When reset: 00$_{16}$

Bit Symbol	Bit Name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN0 0 0 1 : AN1 0 1 0 : AN2 0 1 1 : AN3 1 0 0 : AN4 1 0 1 : AN5 1 1 0 : AN6 1 1 1 : AN7 (Note 2, 3)	O	O
CH1			O	O
CH2			O	O
MD0	A/D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 Repeat sweep mode 1 (Note 2)	O	O
MD1			O	O
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{\text{ADTRG}}$ trigger	O	O
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion enabled (Note 4)	O	O
CKS0	Frequency select bit 0 (Note 5)	0 : f_{AD}/3 or f_{AD}/4 is selected 1 : f_{AD} or f_{AD}/2 is selected	O	O

Note 1: If the AD control regsiter 0 is rewritten during A/D conversion, the conversion result is indeterminate.
Note 2: When changing A/D operation mode, reset the analog input pin.
Note 3: This bit is disabled in single-sweep mode, repeat-sweep mode 0 and repeat-sweep mode 1.
Note 4: Set to "1" when $\overline{\text{ADTRG}}$ is selected.
Note 5: When f(X_{IN}) exceeds 10 MHz, the ϕ AD frequency must be less than 10 MHz by dividing.

AD control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

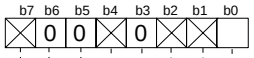
Symbol: ADCON1 — Address: 03D7$_{16}$ — When reset: 00$_{16}$

Bit Symbol	Bit Name	Function	R	W
SCAN0	A/D sweep pin select bit	b1 b0 0 0 : AN0, AN1 (AN0) 0 1 : AN0 to AN3 (AN0, AN1) 1 0 : AN0 to AN5 (AN0 to AN2) 1 1 : AN0 to AN7 (AN0 to AN3) (Note 2)	O	O
SCAN1			O	O
MD2	A/D operation mode select bit 1	0 : Any mode other than repeat-sweep mode 1 1 : Repeat-sweep mode 1	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency select bit 1(Note 3)	0 : fAD/2 or fAD/4 is selected 1 : fAD/1 or fAD/3 is selected	O	O
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	O	O
Reserved		Must always be set to "0"	—	—

Note 1: If the AD control regsiter 1 is rewritten during A/D conversion, the conversion result is indeterminate.
Note 2: This bit is invalid in one-shot mode and repeat mode. Channels shown in parentheses are valid when repeat-sweep mode 1 (bit 2 = "1") is selected.
Note 3: When f(X_{IN}) exceeds 10 MHz, the ϕ AD frequency must be less than 10 MHz by dividing.

Figure 2.9.2. AD converter-related registers (1)

AD control register 2 (Note)

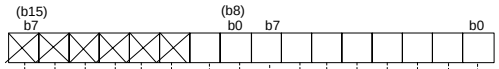


Symbol: ADCON2 | Address: 03D4$_{16}$ | When reset: X00X0XX0$_2$

Bit Symbol	Bit Name	Function	R	W
SMP	A/D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	O	O
Nothing is assigned. Write “0” when writing to these bits. The values are indeterminate when read.			–	–
Reserved		Must always be set to "0"	O	O
Nothing is assigned. Write “0” when writing to this bit. The value is indeterminate when read.			–	–
Reserved		Must always be set to "0"	O	O
Nothing is assigned. Write “0” when writing to this bit. The values are indeterminate when read.			–	–

Note: If the AD control regsiter 2 is rewritten during A/D conversion, the conversion result is indeterminate.

AD register i (i = 0 to 7)



Symbol	Address	When reset
ADi (i = 0 to 2)	03C0$_{16}$ to 3C1$_{16}$, 03C2$_{16}$ to 3C3$_{16}$, 03C4$_{16}$ to 3C5$_{16}$	Indeterminate
ADi (i = 3 to 5)	03C6$_{16}$ to 3C7$_{16}$, 03C8$_{16}$ to 3C9$_{16}$, 03CA$_{16}$ to 3CB$_{16}$	Indeterminate
ADi (i = 6 to 7)	03CC$_{16}$ to 3CD$_{16}$, 03CE$_{16}$ to 3CF$_{16}$	Indeterminate

Function	R	W
Eight low-order bits of A/D conversion results	O	×
During 10-bit mode: Two high-order bits of A/D conversion results During 8-bit mode: The values are indeterminate when read	O	×
Nothing is assigned. Write “0” when writing to these bits. The values are indeterminate when read.	–	–

Figure 2.9.3. A/D converter-related registers (2)

2.9.2 Operation of A/D converter (one-shot mode)

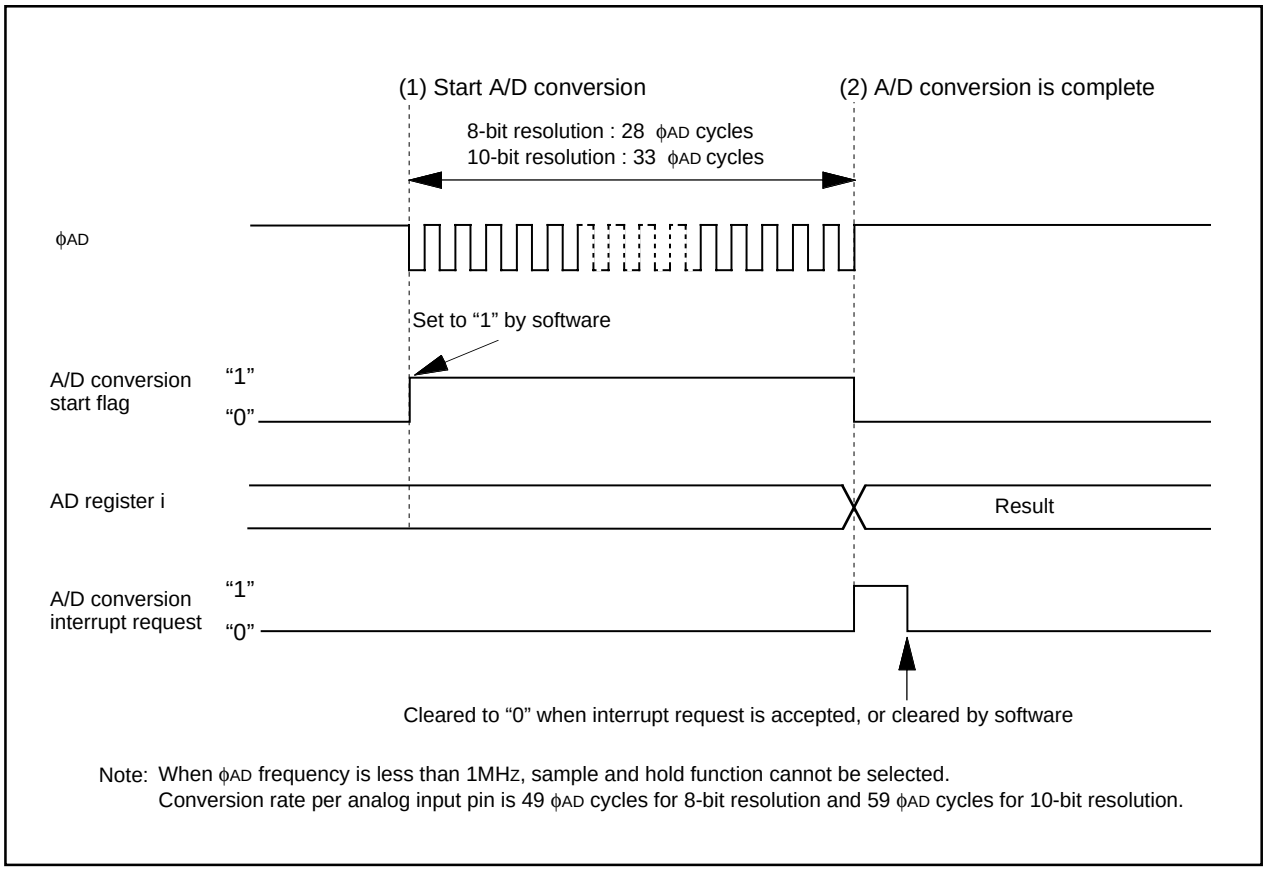
In one-shot mode, choose functions from those listed in Table 2.9.2. Operations of the circled items are described below. Figure 2.9.4 shows the operation timing, and Figure 2.9.5 shows the set-up procedure.

Table 2.9.2. Choosed functions

Item	Set-up	
Operation clock ϕ_{AD}	O	Divided-by-4 f_{AD} / divided-by-3 f_{AD} / divided-by-2 f_{AD} / f_{AD}
Resolution	O	8-bit / 10-bit
Analog input pin	O	One of AN_0 pin to AN_7 pin
Trigger for starting A/D conversion	O	Software trigger
		Trigger by $\overline{ADTRG}$
Sample & Hold		Not activated
	O	Activated

Operation (1) Setting the A/D conversion start flag to “1” causes the A/D converter to begin operating.

(2) After A/D conversion is completed, the content of the successive comparison register (conversion result) is transmitted to AD register i. At this time, the A/D conversion interrupt request bit goes to “1”. Also, the A/D conversion start flag goes to “0”, and the A/D converter stops operating.



Note: When ϕ_{AD} frequency is less than 1MHz, sample and hold function cannot be selected.
Conversion rate per analog input pin is 49 ϕ_{AD} cycles for 8-bit resolution and 59 ϕ_{AD} cycles for 10-bit resolution.

Figure 2.9.4. Operation timing of one-shot mode

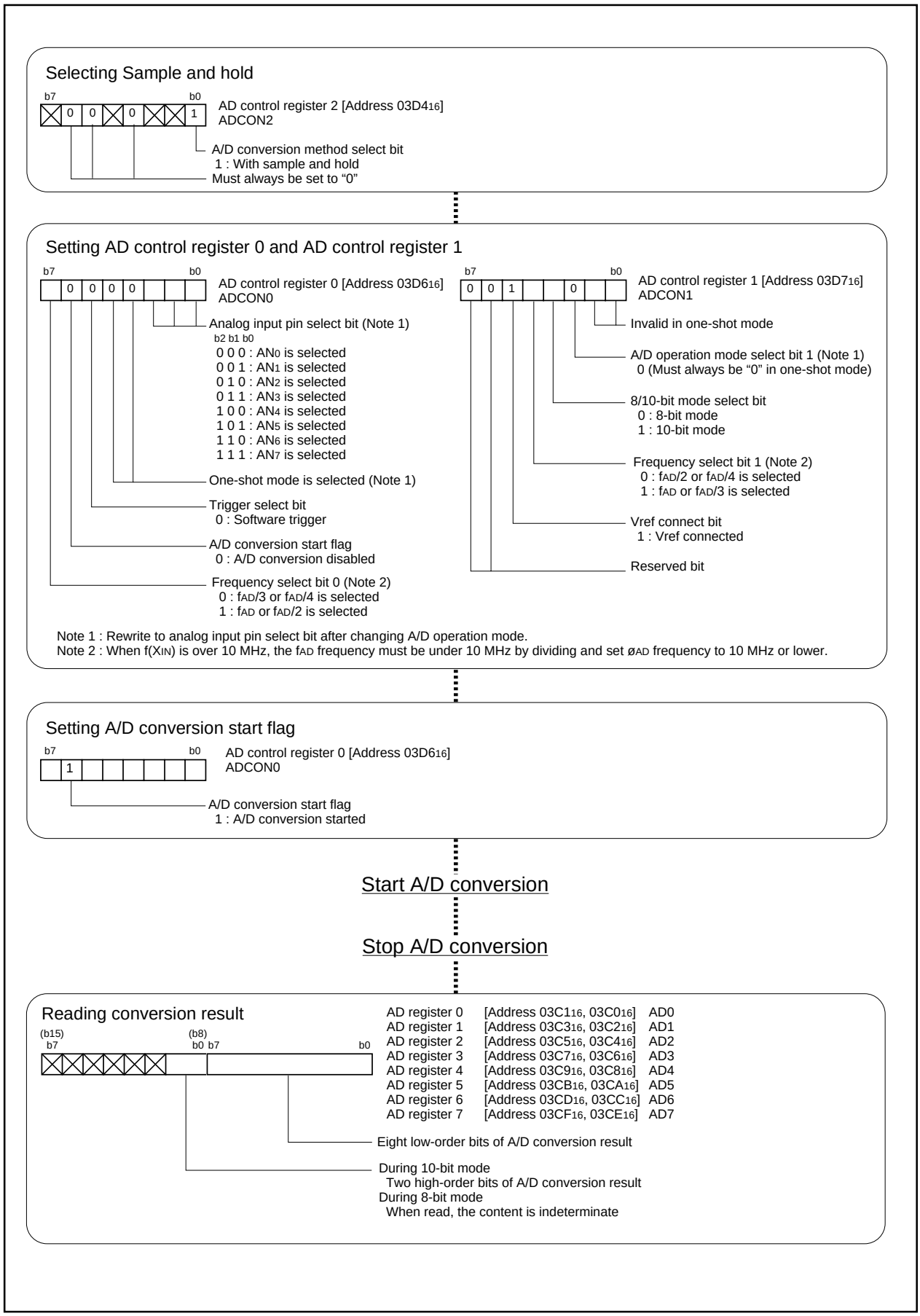

Selecting Sample and hold
b7 b0
0 0 0 1
AD control register 2 [Address 03D416]
ADCON2
A/D conversion method select bit
1 : With sample and hold
Must always be set to "0"
Setting AD control register 0 and AD control register 1
b7 b0
0 0 0 0
AD control register 0 [Address 03D616]
ADCON0
Analog input pin select bit (Note 1)
b2 b1 b0
0 0 0 : AN0 is selected
0 0 1 : AN1 is selected
0 1 0 : AN2 is selected
0 1 1 : AN3 is selected
1 0 0 : AN4 is selected
1 0 1 : AN5 is selected
1 1 0 : AN6 is selected
1 1 1 : AN7 is selected
One-shot mode is selected (Note 1)
Trigger select bit
0 : Software trigger
A/D conversion start flag
0 : A/D conversion disabled
Frequency select bit 0 (Note 2)
0 : fAD/3 or fAD/4 is selected
1 : fAD or fAD/2 is selected
b7 b0
0 0 1 0
AD control register 1 [Address 03D716]
ADCON1
Invalid in one-shot mode
A/D operation mode select bit 1 (Note 1)
0 (Must always be "0" in one-shot mode)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1 (Note 2)
0 : fAD/2 or fAD/4 is selected
1 : fAD or fAD/3 is selected
Vref connect bit
1 : Vref connected
Reserved bit
Note 1 : Rewrite to analog input pin select bit after changing A/D operation mode.
Note 2 : When f(XIN) is over 10 MHz, the fAD frequency must be under 10 MHz by dividing and set øAD frequency to 10 MHz or lower.
Setting A/D conversion start flag
b7 b0
1
AD control register 0 [Address 03D616]
ADCON0
A/D conversion start flag
1 : A/D conversion started
Start A/D conversion
Stop A/D conversion
Reading conversion result
(b15) (b8)
b7 b0 b7 b0
AD register 0 [Address 03C116, 03C016] AD0
AD register 1 [Address 03C316, 03C216] AD1
AD register 2 [Address 03C516, 03C416] AD2
AD register 3 [Address 03C716, 03C616] AD3
AD register 4 [Address 03C916, 03C816] AD4
AD register 5 [Address 03CB16, 03CA16] AD5
AD register 6 [Address 03CD16, 03CC16] AD6
AD register 7 [Address 03CF16, 03CE16] AD7
Eight low-order bits of A/D conversion result
During 10-bit mode
Two high-order bits of A/D conversion result
During 8-bit mode
When read, the content is indeterminate

Figure 2.9.5. Set-up procedure of one-shot mode

2.9.3 Operation of A/D Converter (in one-shot mode, an external trigger selected)

In one-shot mode, choose functions from those listed in Table 2.9.3. Operations of the circled items are described below. Figure 2.9.6 shows timing chart, and Figure 2.9.7 shows the set-up procedure.

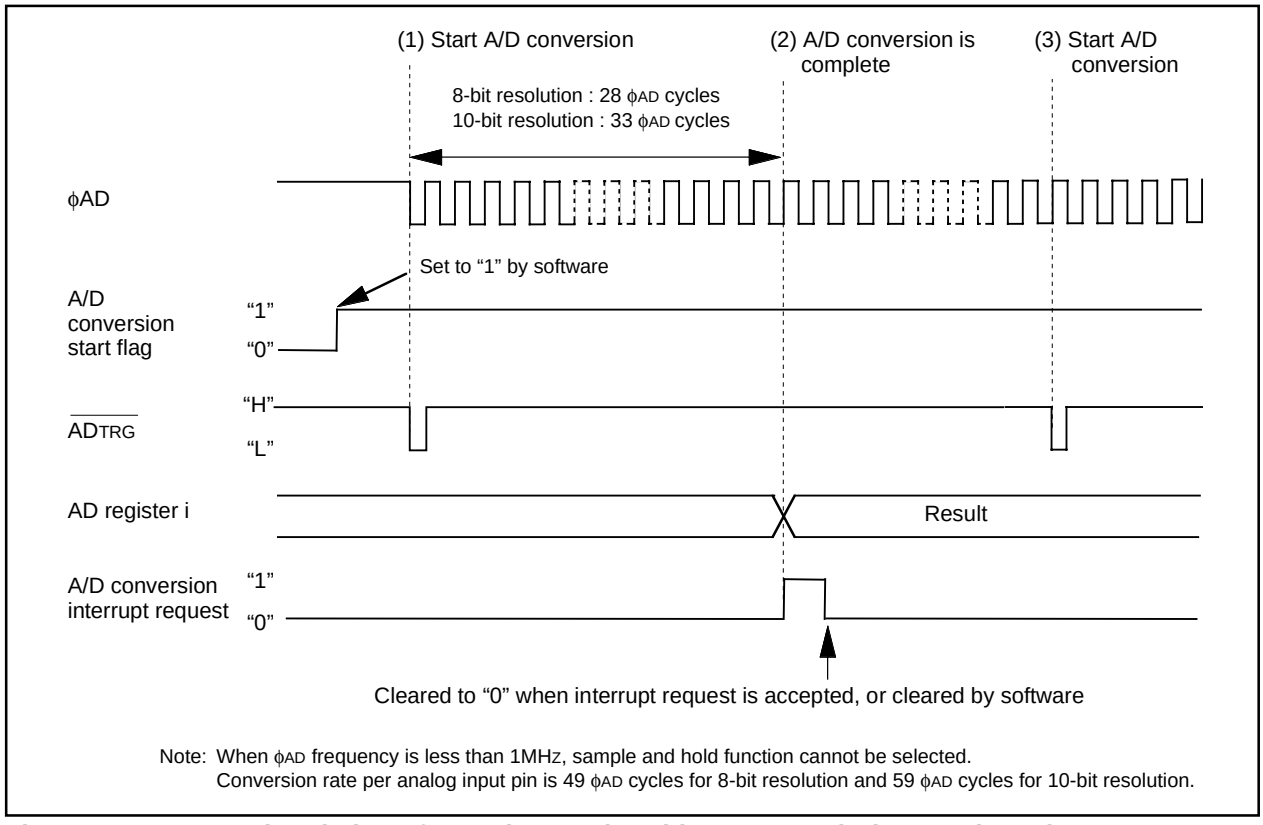
Table 2.9.3. Choosed functions

Item	Set-up	
Operation clock ϕ_{AD}	O	Divided-by-4 f_{AD} / divided-by-3 f_{AD} / divided-by-2 f_{AD} / f_{AD}
Resolution	O	8-bit / 10-bit
Analog input pin	O	One of AN_0 pin to AN_7 pin
Trigger for starting A/D conversion		Software trigger
	O	Trigger by $\overline{ADTRG}$
Sample & Hold		Not activated
	O	Activated

Operation (1) If the level of the $\overline{ADTRG}$ changes from “H” to “L” with the A/D conversion start flag set to “1”, the A/D converter begins operating.

(2) After A/D conversion is completed, the content of the successive comparison register (conversion result) is transmitted to AD register i. At this time, the A/D conversion interrupt request bit goes to “1”. Also the A/D converter stops operating.

(3) If the level of the $\overline{ADTRG}$ pin changes from “H” to “L”, the A/D converter carries out conversion from step (1) again. If the level of the $\overline{ADTRG}$ pin changes from “H” to “L” while conversion is in progress, the A/D converter stops the A/D conversion in process, and carries out conversion from step (1) again.



Note: When ϕ_{AD} frequency is less than 1MHz, sample and hold function cannot be selected.
Conversion rate per analog input pin is 49 ϕ_{AD} cycles for 8-bit resolution and 59 ϕ_{AD} cycles for 10-bit resolution.

Figure 2.9.6. Operation timing of one-shot mode, with an external trigger selected

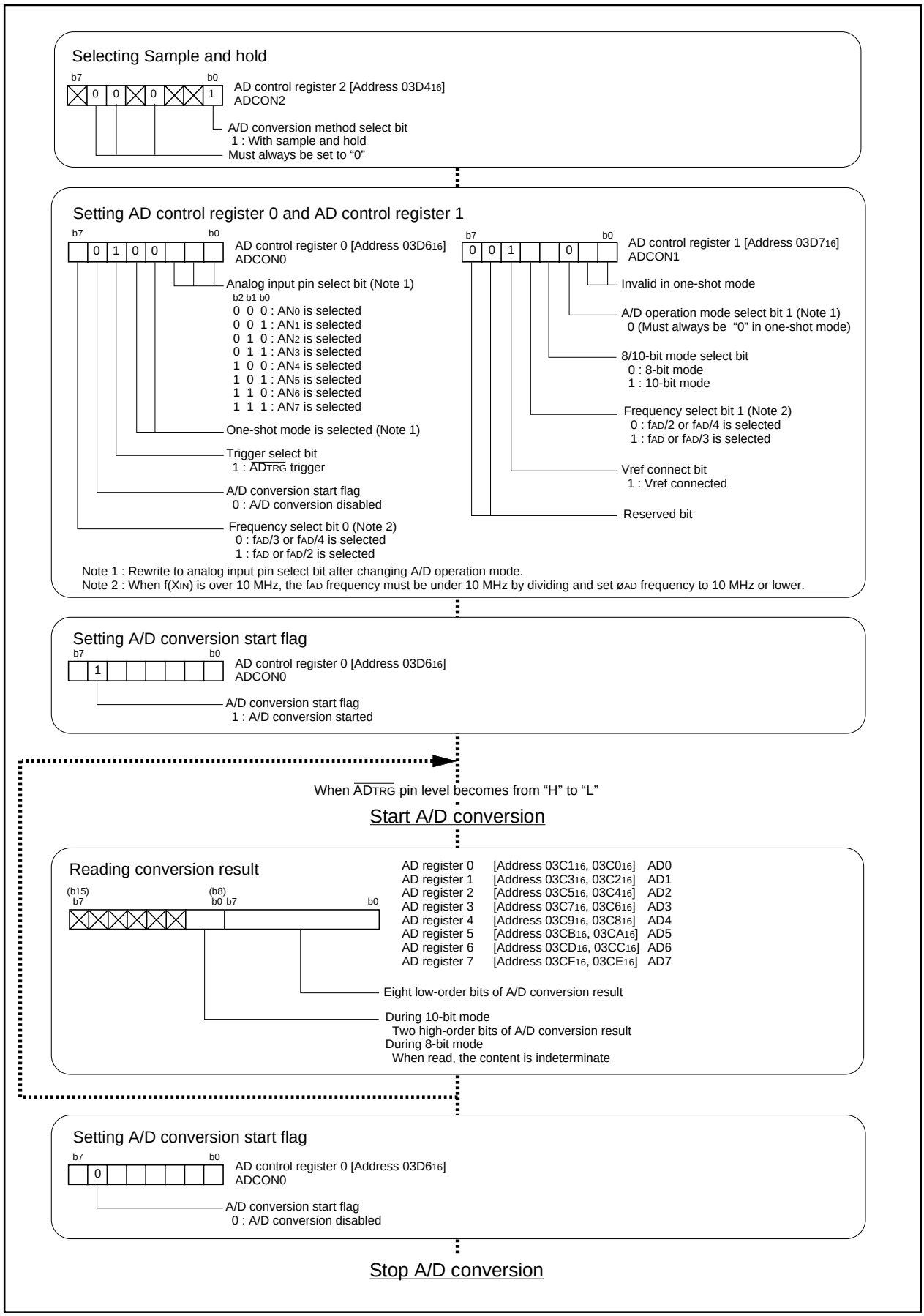

Selecting Sample and hold
b7 b0
0 0 0 1
AD control register 2 [Address 03D416]
ADCON2
A/D conversion method select bit
1 : With sample and hold
Must always be set to "0"
Setting AD control register 0 and AD control register 1
b7 b0
0 1 0 0
AD control register 0 [Address 03D616]
ADCON0
Analog input pin select bit (Note 1)
b2 b1 b0
0 0 0 : AN0 is selected
0 0 1 : AN1 is selected
0 1 0 : AN2 is selected
0 1 1 : AN3 is selected
1 0 0 : AN4 is selected
1 0 1 : AN5 is selected
1 1 0 : AN6 is selected
1 1 1 : AN7 is selected
One-shot mode is selected (Note 1)
Trigger select bit
1 : ADTRG trigger
A/D conversion start flag
0 : A/D conversion disabled
Frequency select bit 0 (Note 2)
0 : fAD/3 or fAD/4 is selected
1 : fAD or fAD/2 is selected
b7 b0
0 0 1 0
AD control register 1 [Address 03D716]
ADCON1
Invalid in one-shot mode
A/D operation mode select bit 1 (Note 1)
0 (Must always be "0" in one-shot mode)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1 (Note 2)
0 : fAD/2 or fAD/4 is selected
1 : fAD or fAD/3 is selected
Vref connect bit
1 : Vref connected
Reserved bit
Note 1 : Rewrite to analog input pin select bit after changing A/D operation mode.
Note 2 : When f(XIN) is over 10 MHz, the fAD frequency must be under 10 MHz by dividing and set øAD frequency to 10 MHz or lower.
Setting A/D conversion start flag
b7 b0
1
AD control register 0 [Address 03D616]
ADCON0
A/D conversion start flag
1 : A/D conversion started
When ADTRG pin level becomes from "H" to "L"
Start A/D conversion
Reading conversion result
(b15) (b8)
b7 b0 b7 b0
AD register 0 [Address 03C116, 03C016] AD0
AD register 1 [Address 03C316, 03C216] AD1
AD register 2 [Address 03C516, 03C416] AD2
AD register 3 [Address 03C716, 03C616] AD3
AD register 4 [Address 03C916, 03C816] AD4
AD register 5 [Address 03CB16, 03CA16] AD5
AD register 6 [Address 03CD16, 03CC16] AD6
AD register 7 [Address 03CF16, 03CE16] AD7
Eight low-order bits of A/D conversion result
During 10-bit mode
Two high-order bits of A/D conversion result
During 8-bit mode
When read, the content is indeterminate
Setting A/D conversion start flag
b7 b0
0
AD control register 0 [Address 03D616]
ADCON0
A/D conversion start flag
0 : A/D conversion disabled
Stop A/D conversion

Figure 2.9.7. Set-up procedure of one-shot mode, with an external trigger selected

2.9.4 Operation of A/D Converter (in repeat mode)

In repeat mode, choose functions from those listed in Table 2.9.4. Operations of the circled items are described below. Figure 2.9.8 shows timing chart, and Figure 2.9.9 shows the set-up procedure.

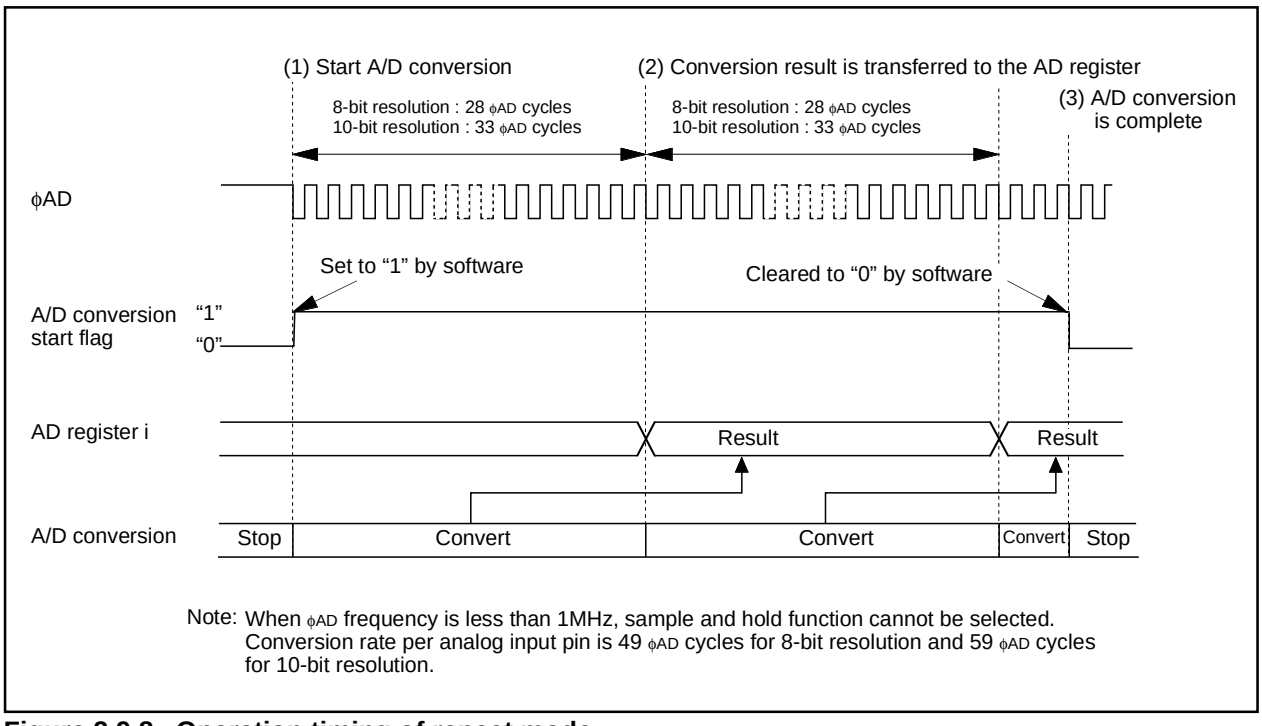
Table 2.9.4. Choosed functions

Item	Set-up	
Operation clock ϕ_{AD}	O	Divided-by-4 f_{AD} / divided-by-3 f_{AD} / divided-by-2 f_{AD} / f_{AD}
Resolution	O	8-bit / 10-bit
Analog input pin	O	One of AN_0 pin to AN_7 pin
Trigger for starting A/D conversion	O	Software trigger
		Trigger by $\overline{ADTRG}$
Sample & Hold		Not activated
	O	Activated

Operation (1) Setting the A/D conversion start flag to “1” causes the A/D converter to start operating.

(2) After the first conversion is completed, the content of the successive comparison register (conversion result) is transmitted to AD register i. The A/D conversion interrupt request bit does not go to “1”.

(3) The A/D converter continues operating until the A/D conversion start flag is set to “0” by software. The conversion result is transmitted to AD register i every time a conversion is completed.



Note: When ϕ_{AD} frequency is less than 1MHz, sample and hold function cannot be selected. Conversion rate per analog input pin is 49 ϕ_{AD} cycles for 8-bit resolution and 59 ϕ_{AD} cycles for 10-bit resolution.

Figure 2.9.8. Operation timing of repeat mode

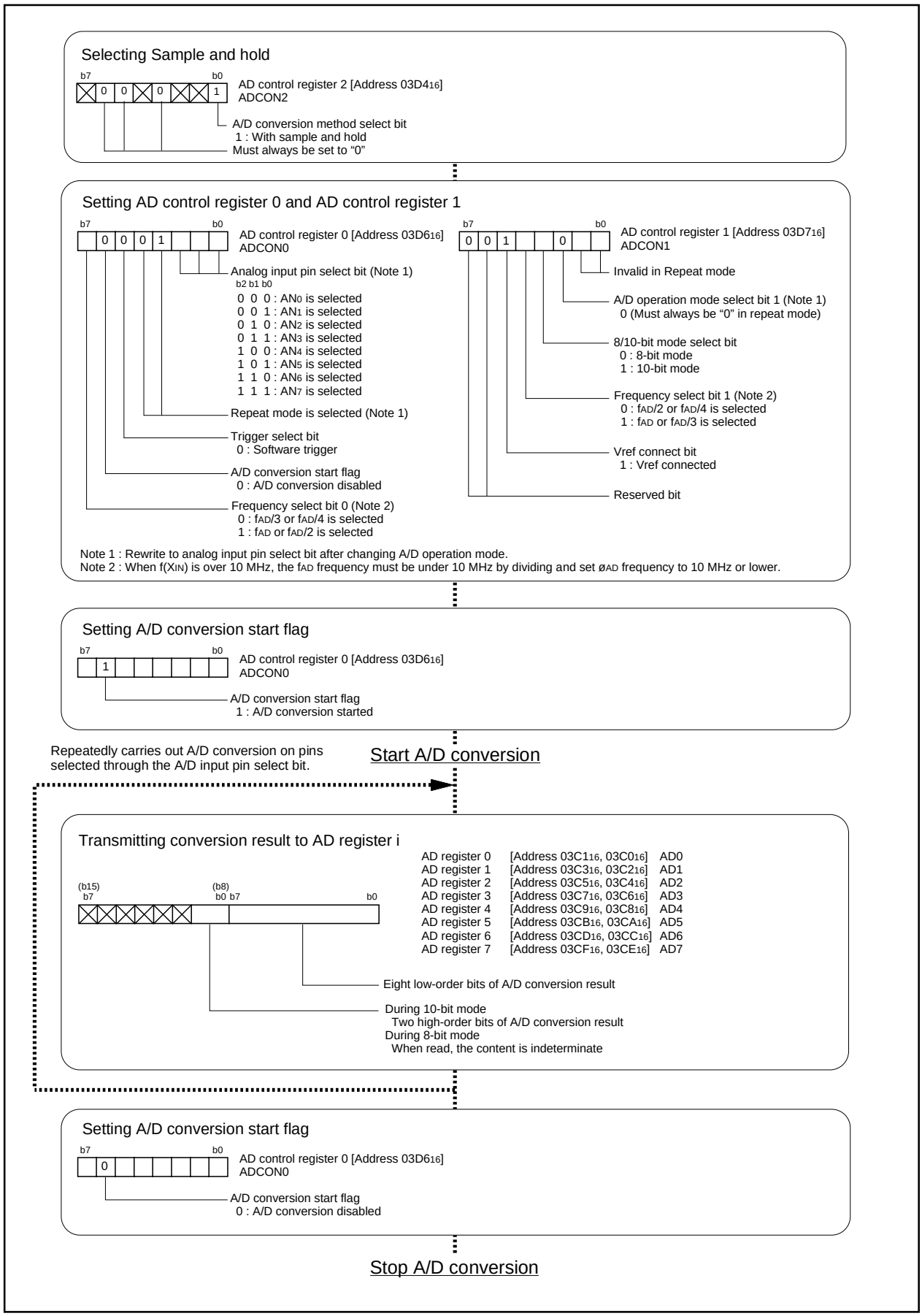

Selecting Sample and hold
b7 b0
0 0 0 1
AD control register 2 [Address 03D416]
ADCON2
A/D conversion method select bit
1 : With sample and hold
Must always be set to "0"
Setting AD control register 0 and AD control register 1
b7 b0
0 0 0 1
AD control register 0 [Address 03D616]
ADCON0
Analog input pin select bit (Note 1)
b2 b1 b0
0 0 0 : AN0 is selected
0 0 1 : AN1 is selected
0 1 0 : AN2 is selected
0 1 1 : AN3 is selected
1 0 0 : AN4 is selected
1 0 1 : AN5 is selected
1 1 0 : AN6 is selected
1 1 1 : AN7 is selected
Repeat mode is selected (Note 1)
Trigger select bit
0 : Software trigger
A/D conversion start flag
0 : A/D conversion disabled
Frequency select bit 0 (Note 2)
0 : fAD/3 or fAD/4 is selected
1 : fAD or fAD/2 is selected
b7 b0
0 0 1 0
AD control register 1 [Address 03D716]
ADCON1
Invalid in Repeat mode
A/D operation mode select bit 1 (Note 1)
0 (Must always be "0" in repeat mode)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1 (Note 2)
0 : fAD/2 or fAD/4 is selected
1 : fAD or fAD/3 is selected
Vref connect bit
1 : Vref connected
Reserved bit
Note 1 : Rewrite to analog input pin select bit after changing A/D operation mode.
Note 2 : When f(XIN) is over 10 MHz, the fAD frequency must be under 10 MHz by dividing and set øAD frequency to 10 MHz or lower.
Setting A/D conversion start flag
b7 b0
1
AD control register 0 [Address 03D616]
ADCON0
A/D conversion start flag
1 : A/D conversion started
Repeatedly carries out A/D conversion on pins selected through the A/D input pin select bit.
Start A/D conversion
Transmitting conversion result to AD register i
(b15) (b8)
b7 b0 b7 b0
AD register 0 [Address 03C116, 03C016] AD0
AD register 1 [Address 03C316, 03C216] AD1
AD register 2 [Address 03C516, 03C416] AD2
AD register 3 [Address 03C716, 03C616] AD3
AD register 4 [Address 03C916, 03C816] AD4
AD register 5 [Address 03CB16, 03CA16] AD5
AD register 6 [Address 03CD16, 03CC16] AD6
AD register 7 [Address 03CF16, 03CE16] AD7
Eight low-order bits of A/D conversion result
During 10-bit mode
Two high-order bits of A/D conversion result
During 8-bit mode
When read, the content is indeterminate
Setting A/D conversion start flag
b7 b0
0
AD control register 0 [Address 03D616]
ADCON0
A/D conversion start flag
0 : A/D conversion disabled
Stop A/D conversion

Figure 2.9.9. Set-up procedure of repeat mode

2.9.5 Operation of A/D Converter (in single sweep mode)

In single sweep mode, choose functions from those listed in Table 2.9.5. Operations of the circled items are described below. Figure 2.9.10 shows timing chart, and Figure 2.9.11 shows the set-up procedure.

Table 2.9.5. Choosed functions

Item		Set-up	Item		Set-up
Operation clock f_{AD}	O	Divided-by-4 f_{AD} / divided-by-3 f_{AD} / divided-by-2 f_{AD} / f_{AD}	Trigger for starting A/D conversion	O	Software trigger
					Trigger by $\overline{ADTRG}$
Resolution	O	8-bit / 10-bit	Sample & Hold		Not activated
Analog input pin	O	AN_0 and AN_1 (2 pins) / AN_0 to AN_3 (4 pins) / AN_0 to AN_5 (6 pins) / AN_0 to AN_7 (8 pins)		O	Activated

Operation (1) Setting the A/D conversion start flag to “1” causes the A/D converter to start the conversion on voltage input to the AN_0 pin.

(2) After the A/D conversion of voltage input to the AN_0 pin is completed, the content of the successive comparison register (conversion result) is transmitted to AD register 0. The A/D converter converts all analog input pins selected by the user. The conversion result is transmitted to AD register i corresponding to each pin, every time conversion on one pin is completed.

(3) When the A/D conversion on all the analog input pins selected is completed, the A/D conversion interrupt request bit goes to “1”. At this time, the A/D conversion start flag goes to “0”. The A/D converter stops operating.

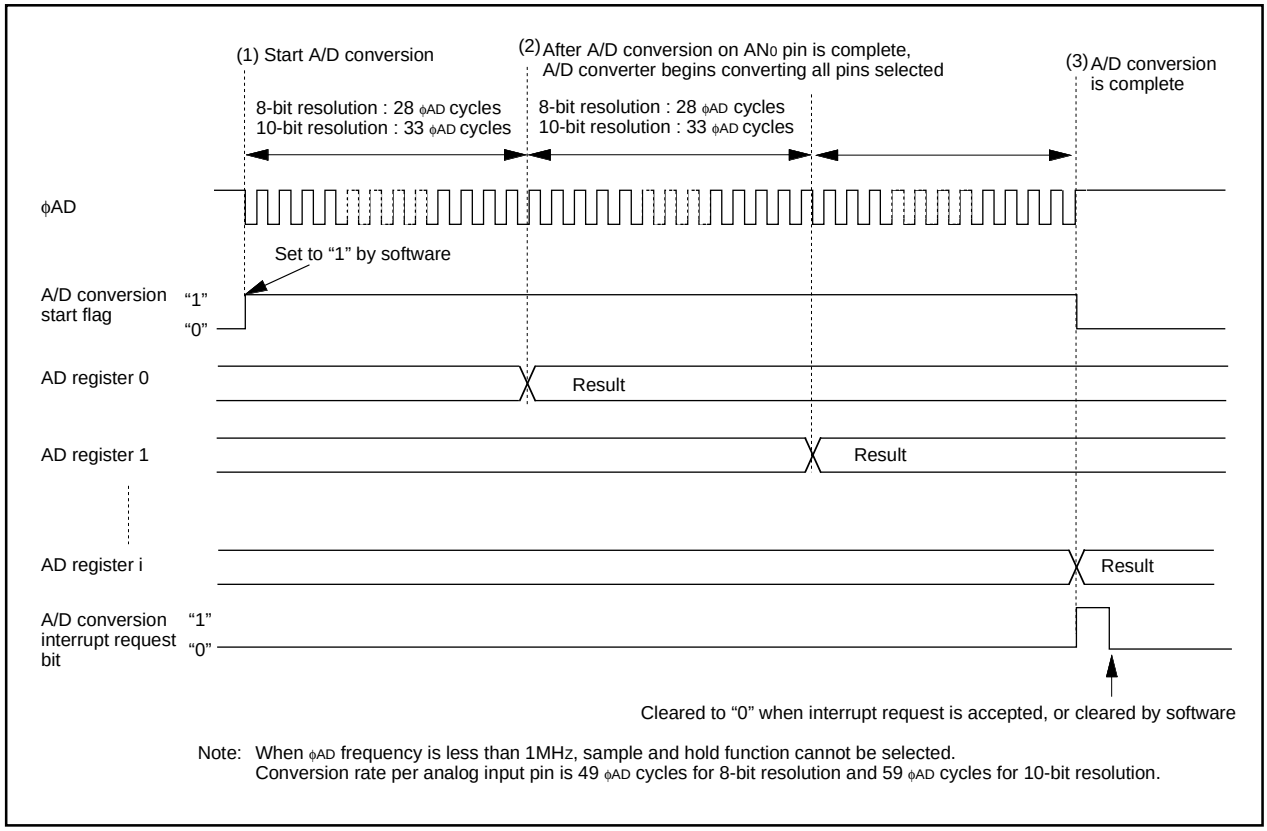

(1) Start A/D conversion
(2) After A/D conversion on AN0 pin is complete, A/D converter begins converting all pins selected
(3) A/D conversion is complete
8-bit resolution : 28 φAD cycles
10-bit resolution : 33 φAD cycles
8-bit resolution : 28 φAD cycles
10-bit resolution : 33 φAD cycles
φAD
Set to “1” by software
A/D conversion start flag
“1”
“0”
AD register 0
Result
AD register 1
Result
AD register i
Result
A/D conversion interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software
Note: When φAD frequency is less than 1MHz, sample and hold function cannot be selected.
Conversion rate per analog input pin is 49 φAD cycles for 8-bit resolution and 59 φAD cycles for 10-bit resolution.

Figure 2.9.10. Operation timing of single sweep mode

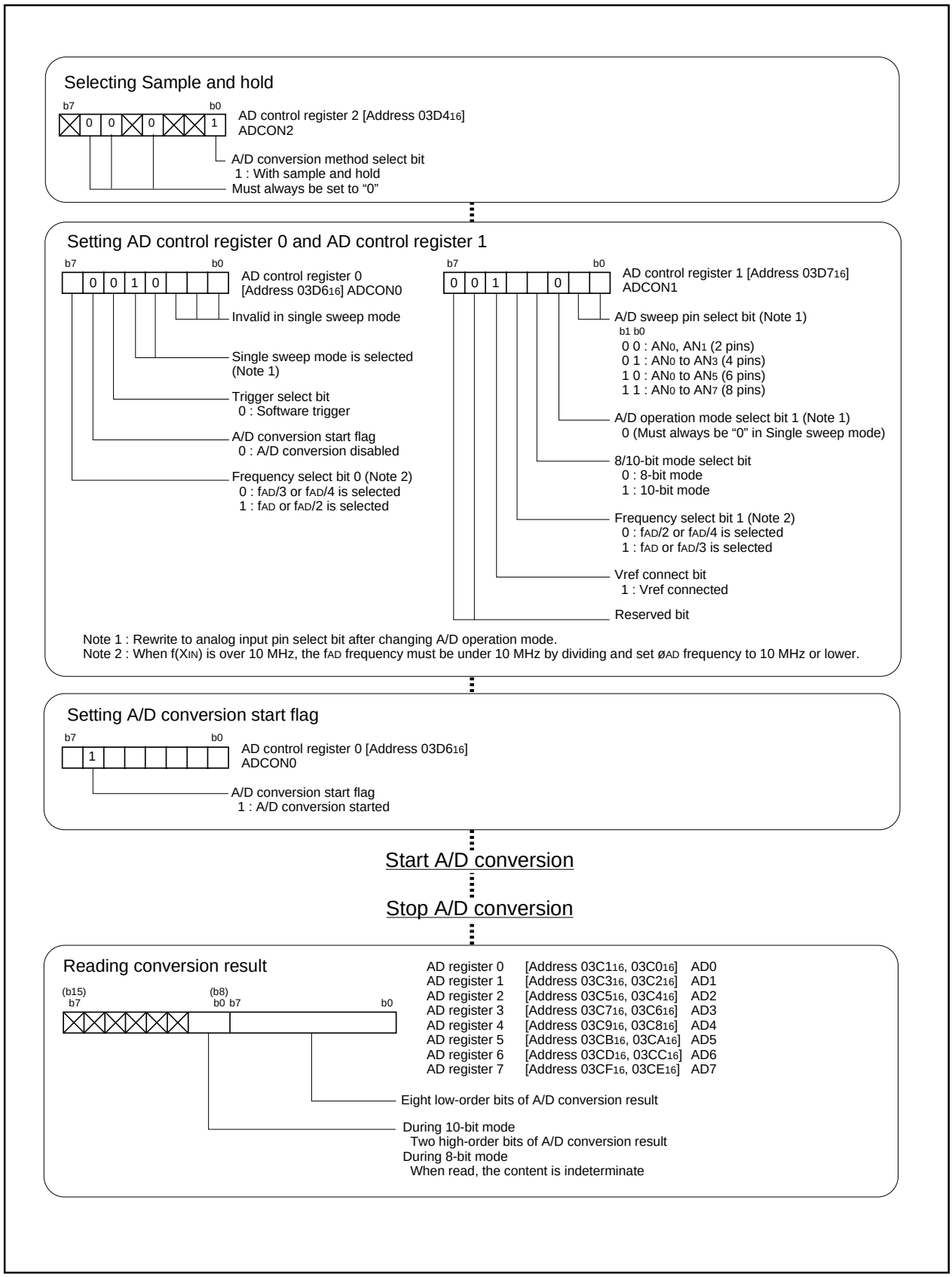

Selecting Sample and hold
b7 b0
0 0 0 1
AD control register 2 [Address 03D416]
ADCON2
A/D conversion method select bit
1 : With sample and hold
Must always be set to "0"
Setting AD control register 0 and AD control register 1
b7 b0
0 0 1 0
AD control register 0
[Address 03D616] ADCON0
Invalid in single sweep mode
Single sweep mode is selected
(Note 1)
Trigger select bit
0 : Software trigger
A/D conversion start flag
0 : A/D conversion disabled
Frequency select bit 0 (Note 2)
0 : fAD/3 or fAD/4 is selected
1 : fAD or fAD/2 is selected
b7 b0
0 0 1 0
AD control register 1 [Address 03D716]
ADCON1
A/D sweep pin select bit (Note 1)
b1 b0
0 0 : AN0, AN1 (2 pins)
0 1 : AN0 to AN3 (4 pins)
1 0 : AN0 to AN5 (6 pins)
1 1 : AN0 to AN7 (8 pins)
A/D operation mode select bit 1 (Note 1)
0 (Must always be "0" in Single sweep mode)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1 (Note 2)
0 : fAD/2 or fAD/4 is selected
1 : fAD or fAD/3 is selected
Vref connect bit
1 : Vref connected
Reserved bit
Note 1 : Rewrite to analog input pin select bit after changing A/D operation mode.
Note 2 : When f(XIN) is over 10 MHz, the fAD frequency must be under 10 MHz by dividing and set øAD frequency to 10 MHz or lower.
Setting A/D conversion start flag
b7 b0
1
AD control register 0 [Address 03D616]
ADCON0
A/D conversion start flag
1 : A/D conversion started
Start A/D conversion
Stop A/D conversion
Reading conversion result
(b15) (b8)
b7 b0 b7 b0
AD register 0 [Address 03C116, 03C016] AD0
AD register 1 [Address 03C316, 03C216] AD1
AD register 2 [Address 03C516, 03C416] AD2
AD register 3 [Address 03C716, 03C616] AD3
AD register 4 [Address 03C916, 03C816] AD4
AD register 5 [Address 03CB16, 03CA16] AD5
AD register 6 [Address 03CD16, 03CC16] AD6
AD register 7 [Address 03CF16, 03CE16] AD7
Eight low-order bits of A/D conversion result
During 10-bit mode
Two high-order bits of A/D conversion result
During 8-bit mode
When read, the content is indeterminate

Figure 2.9.11. Set-up procedure of single sweep mode

2.9.6 Operation of A/D Converter (in repeat sweep mode 0)

In repeat sweep 0 mode, choose functions from those listed in Table 2.9.6. Operations of the circled items are described below. Figure 2.9.12 shows timing chart, and Figure 2.9.13 shows the set-up procedure.

Table 2.9.6. Choosed functions

Item	Set-up		Item	Set-up	
Operation clock f_{AD}	**O**	Divided-by-4 f_{AD} / divided-by-3 f_{AD} / divided-by-2 f_{AD} / f_{AD}	Trigger for starting A/D conversion	**O**	Software trigger
					Trigger by $\overline{ADTRG}$
Resolution	**O**	8-bit / 10-bit	Sample & Hold		Not activated
Analog input pin	**O**	AN_0 and AN_1 (2 pins) / AN_0 to AN_3 (4 pins) / AN_0 to AN_5 (6 pins) / AN_0 to AN_7 (8 pins)		**O**	Activated

Operation (1) Setting the A/D conversion start flag to “1” causes the A/D converter to start the conversion on voltage input to the AN_0 pin.

(2) After the A/D conversion of voltage input to the AN_0 pin is completed, the content of the successive comparison register (conversion result) is transmitted to AD register 0.

(3) The A/D converter converts all pins selected by the user. The conversion result is transmitted to AD register i corresponding to each pin every time A/D conversion on the pin is completed. The A/D conversion interrupt request bit does not go to “1”.

(4) The A/D converter continues operating until the A/D conversion start flag is set to “0” by software.

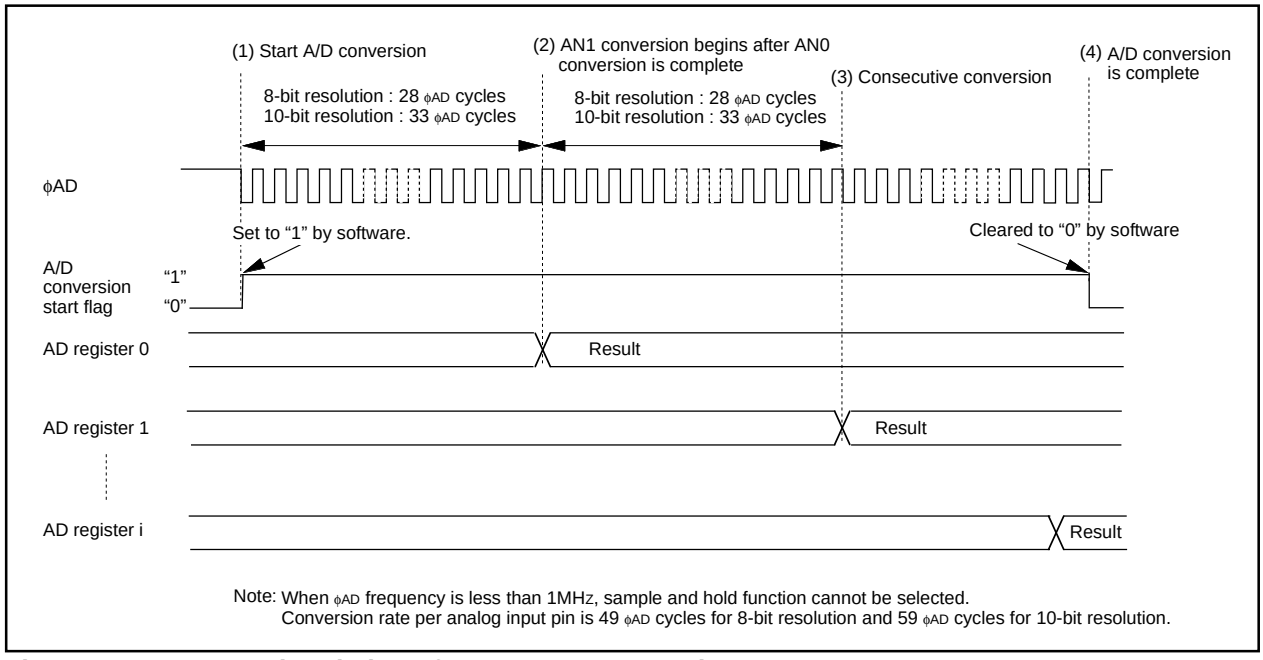

(1) Start A/D conversion
(2) AN1 conversion begins after AN0 conversion is complete
(3) Consecutive conversion
(4) A/D conversion is complete
8-bit resolution : 28 ϕAD cycles
10-bit resolution : 33 ϕAD cycles
8-bit resolution : 28 ϕAD cycles
10-bit resolution : 33 ϕAD cycles
ϕAD
Set to “1” by software.
Cleared to “0” by software
A/D conversion start flag
“1”
“0”
AD register 0
Result
AD register 1
Result
AD register i
Result
Note: When ϕAD frequency is less than 1MHz, sample and hold function cannot be selected.
Conversion rate per analog input pin is 49 ϕAD cycles for 8-bit resolution and 59 ϕAD cycles for 10-bit resolution.

Figure 2.9.12. Operation timing of repeat sweep 0 mode

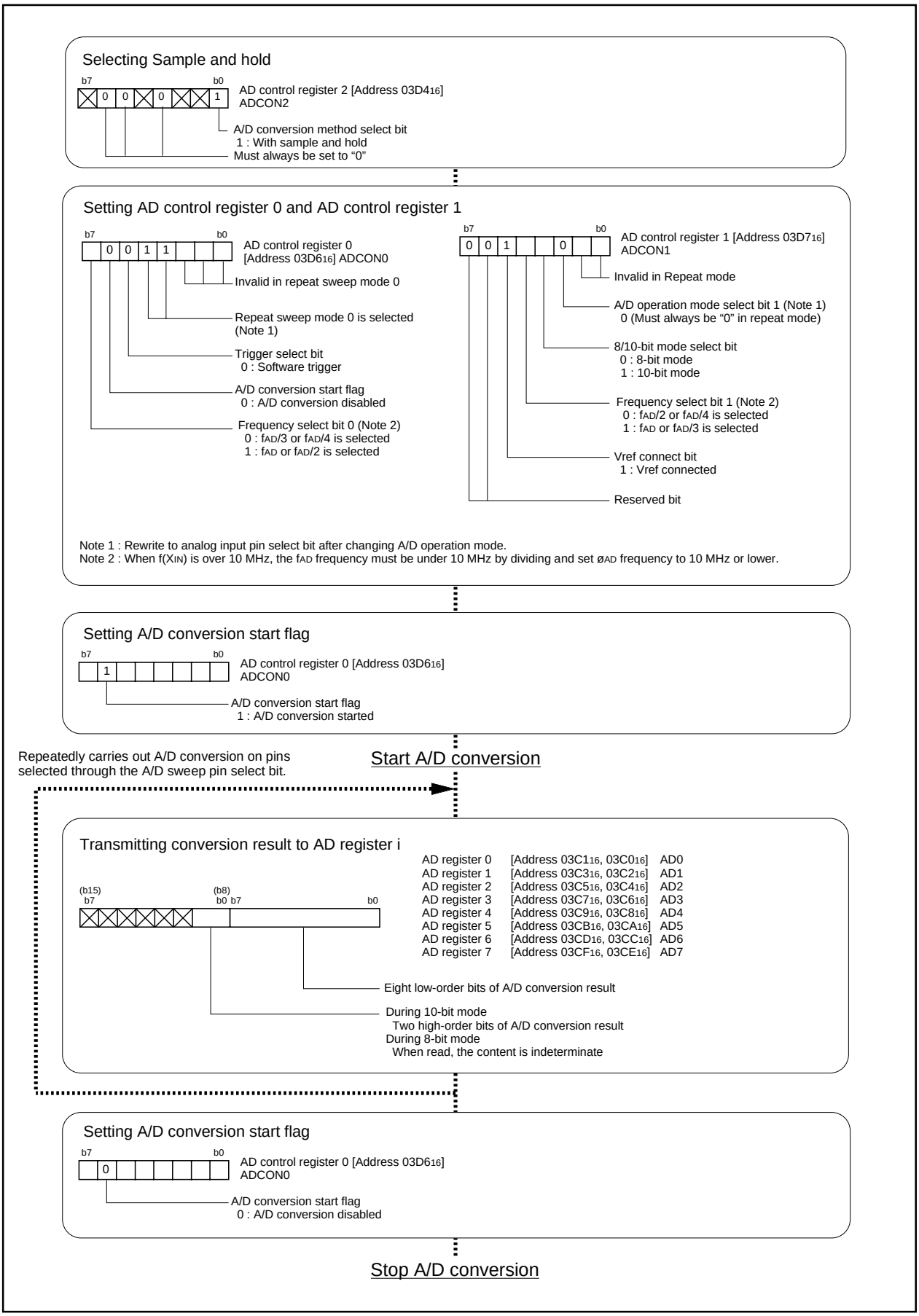

Selecting Sample and hold
b7 b0
0 0 0 1
AD control register 2 [Address 03D416]
ADCON2
A/D conversion method select bit
1 : With sample and hold
Must always be set to "0"
Setting AD control register 0 and AD control register 1
b7 b0
0 0 1 1
AD control register 0
[Address 03D616] ADCON0
Invalid in repeat sweep mode 0
Repeat sweep mode 0 is selected
(Note 1)
Trigger select bit
0 : Software trigger
A/D conversion start flag
0 : A/D conversion disabled
Frequency select bit 0 (Note 2)
0 : fAD/3 or fAD/4 is selected
1 : fAD or fAD/2 is selected
b7 b0
0 0 1 0
AD control register 1 [Address 03D716]
ADCON1
Invalid in Repeat mode
A/D operation mode select bit 1 (Note 1)
0 (Must always be "0" in repeat mode)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1 (Note 2)
0 : fAD/2 or fAD/4 is selected
1 : fAD or fAD/3 is selected
Vref connect bit
1 : Vref connected
Reserved bit
Note 1 : Rewrite to analog input pin select bit after changing A/D operation mode.
Note 2 : When f(XIN) is over 10 MHz, the fAD frequency must be under 10 MHz by dividing and set øAD frequency to 10 MHz or lower.
Setting A/D conversion start flag
b7 b0
1
AD control register 0 [Address 03D616]
ADCON0
A/D conversion start flag
1 : A/D conversion started
Start A/D conversion
Repeatedly carries out A/D conversion on pins
selected through the A/D sweep pin select bit.
Transmitting conversion result to AD register i
AD register 0 [Address 03C116, 03C016] AD0
AD register 1 [Address 03C316, 03C216] AD1
AD register 2 [Address 03C516, 03C416] AD2
AD register 3 [Address 03C716, 03C616] AD3
AD register 4 [Address 03C916, 03C816] AD4
AD register 5 [Address 03CB16, 03CA16] AD5
AD register 6 [Address 03CD16, 03CC16] AD6
AD register 7 [Address 03CF16, 03CE16] AD7
(b15) (b8)
b7 b0 b7 b0
Eight low-order bits of A/D conversion result
During 10-bit mode
Two high-order bits of A/D conversion result
During 8-bit mode
When read, the content is indeterminate
Setting A/D conversion start flag
b7 b0
0
AD control register 0 [Address 03D616]
ADCON0
A/D conversion start flag
0 : A/D conversion disabled
Stop A/D conversion

Figure 2.9.13. Set-up procedure of repeat sweep 0 mode

2.9.7 Operation of A/D Converter (in repeat sweep mode 1)

In repeat sweep 1 mode, choose functions from those listed in Table 2.9.7. Operations of the circled items are described below. Figure 2.9.14 shows ANi pin's sweep sequence, Figure 2.9.15 shows timing chart, and Figure 2.9.16 shows the set-up procedure.

Table 2.9.7. Choosed functions

Item	Set-up		Item	Set-up	
Operation clock f_{AD}	O	Divided-by-4 f_{AD} / divided-by-3 f_{AD} / divided-by-2 f_{AD} / f_{AD}	Trigger for starting A/D conversion	O	Software trigger
					Trigger by $\overline{ADTRG}$
Resolution	O	8-bit / 10-bit	Sample & Hold		Not activated
Analog input pin	O	AN_0 (1 pin) / AN_0 and AN_1 (2 pins) / AN_0 to AN_2 (3 pins) / AN_0 to AN_3 (4 pins)		O	Activated

Operation (1) Setting the A/D conversion start flag to “1” causes the A/D converter to start the conversion on voltage input to the AN_0 pin.

(2) After the A/D conversion on voltage input to the AN_0 pin is completed, the content of the successive comparison register (conversion result) is transmitted to AD register 0.

(3) Every time the A/D converter carries out A/D conversion on a selected analog input pin, the A/D converter carries out A/D conversion on only one unselected pin, and then the A/D converter carries out A/D conversion from the AN_0 pin again. The conversion result is transmitted to the corresponding AD register i every time conversion on a pin is completed. The A/D conversion interrupt request bit does not go to “1”.

(4) The A/D converter continues operating until software goes the A/D conversion start flag to “0”.

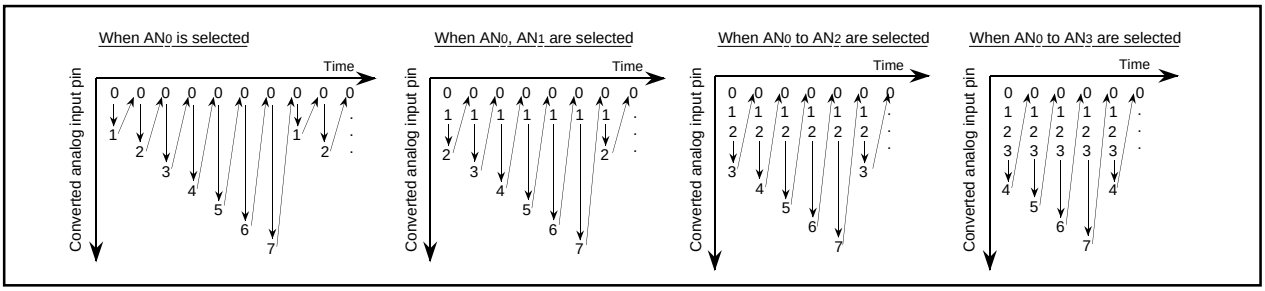
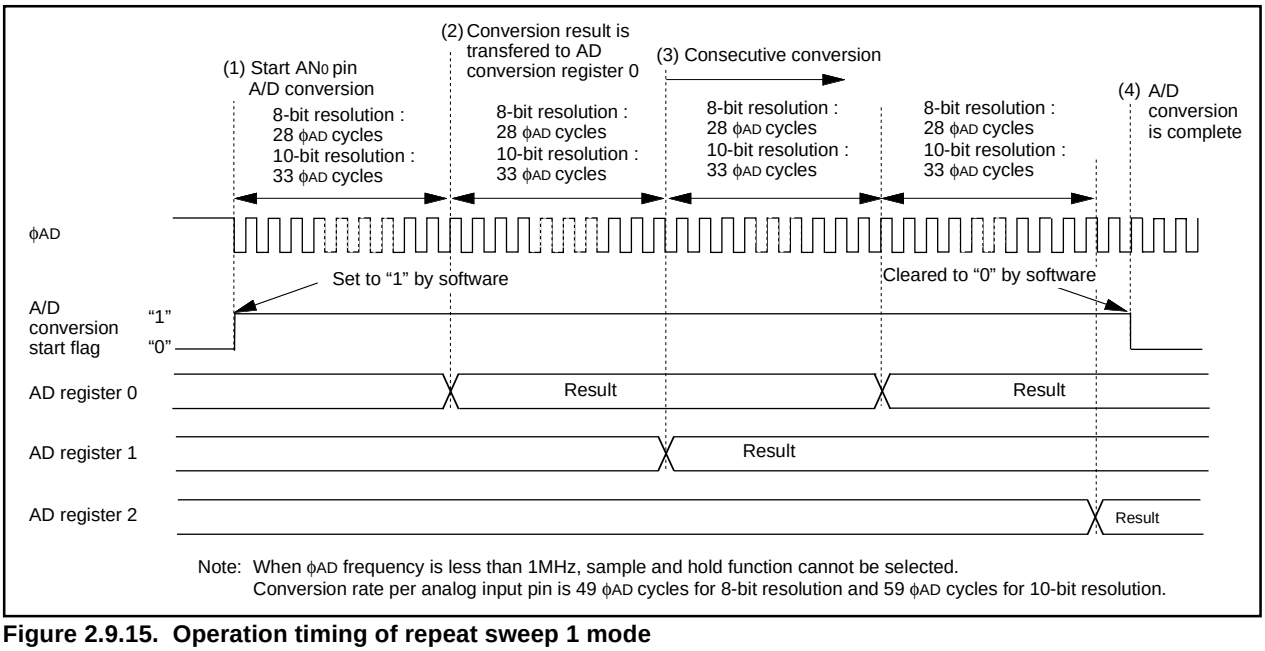

When AN0 is selected
When AN0, AN1 are selected
When AN0 to AN2 are selected
When AN0 to AN3 are selected
Converted analog input pin
Time

Figure 2.9.14. ANi pin's sweep sequence in repeat sweep mode



Note: When ϕ_{AD} frequency is less than 1MHz, sample and hold function cannot be selected.
Conversion rate per analog input pin is 49 ϕ_{AD} cycles for 8-bit resolution and 59 ϕ_{AD} cycles for 10-bit resolution.

Figure 2.9.15. Operation timing of repeat sweep 1 mode

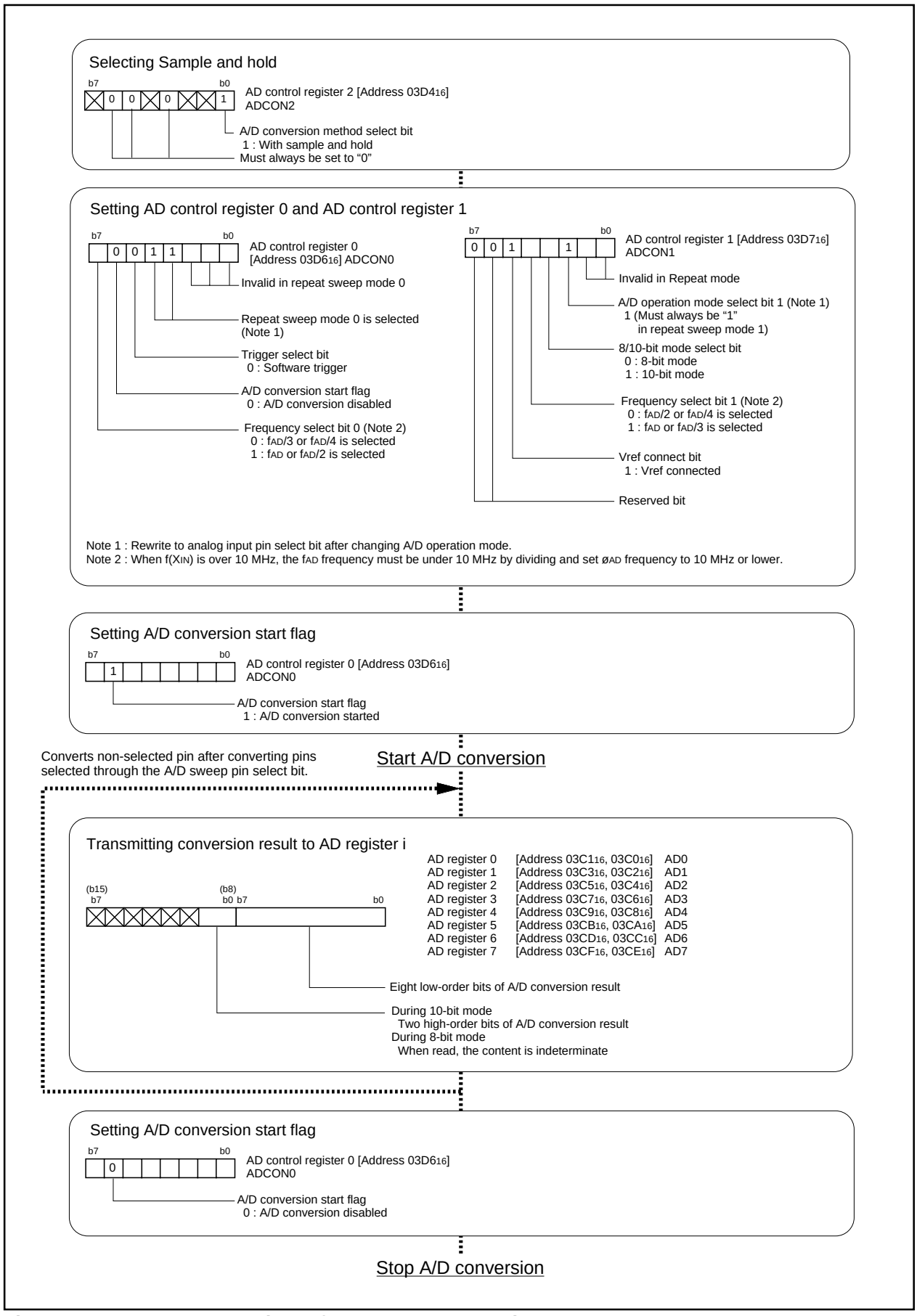

Selecting Sample and hold
b7 b0
0 0 0 1
AD control register 2 [Address 03D416]
ADCON2
A/D conversion method select bit
1 : With sample and hold
Must always be set to "0"
Setting AD control register 0 and AD control register 1
b7 b0
0 0 1 1
AD control register 0
[Address 03D616] ADCON0
Invalid in repeat sweep mode 0
Repeat sweep mode 0 is selected
(Note 1)
Trigger select bit
0 : Software trigger
A/D conversion start flag
0 : A/D conversion disabled
Frequency select bit 0 (Note 2)
0 : fAD/3 or fAD/4 is selected
1 : fAD or fAD/2 is selected
b7 b0
0 0 1 1
AD control register 1 [Address 03D716]
ADCON1
Invalid in Repeat mode
A/D operation mode select bit 1 (Note 1)
1 (Must always be "1"
in repeat sweep mode 1)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1 (Note 2)
0 : fAD/2 or fAD/4 is selected
1 : fAD or fAD/3 is selected
Vref connect bit
1 : Vref connected
Reserved bit
Note 1 : Rewrite to analog input pin select bit after changing A/D operation mode.
Note 2 : When f(XIN) is over 10 MHz, the fAD frequency must be under 10 MHz by dividing and set øAD frequency to 10 MHz or lower.
Setting A/D conversion start flag
b7 b0
1
AD control register 0 [Address 03D616]
ADCON0
A/D conversion start flag
1 : A/D conversion started
Converts non-selected pin after converting pins selected through the A/D sweep pin select bit.
Start A/D conversion
Transmitting conversion result to AD register i
(b15) (b8)
b7 b0 b7 b0
AD register 0 [Address 03C116, 03C016] AD0
AD register 1 [Address 03C316, 03C216] AD1
AD register 2 [Address 03C516, 03C416] AD2
AD register 3 [Address 03C716, 03C616] AD3
AD register 4 [Address 03C916, 03C816] AD4
AD register 5 [Address 03CB16, 03CA16] AD5
AD register 6 [Address 03CD16, 03CC16] AD6
AD register 7 [Address 03CF16, 03CE16] AD7
Eight low-order bits of A/D conversion result
During 10-bit mode
Two high-order bits of A/D conversion result
During 8-bit mode
When read, the content is indeterminate
Setting A/D conversion start flag
b7 b0
0
AD control register 0 [Address 03D616]
ADCON0
A/D conversion start flag
0 : A/D conversion disabled
Stop A/D conversion

Figure 2.9.16. Set-up procedure of repeat sweep 1 mode

2.9.8 Precautions for A/D Converter

(1) Write to each bit (except bit 6) of AD control register 0, to each bit of AD control register 1, and to bit 0 of AD control register 2 when A/D conversion is stopped (before a trigger occurs). In particular, when the Vref connection bit is changed from 0 to 1, start A/D conversion after an elapse of 1 µs or longer.

(2) To reduce conversion error due to noise, connect a voltage to the AVcc pin and to the Vref pin from an independent source. It is recommended to connect a capacitor between the AVss pin and the AVcc pin, between the AVss pin and the Vref pin, and between the AVss pin and the analog input pin (ANi). Figure 2.9.17 shows the an example of connecting the capacitors to these pins.

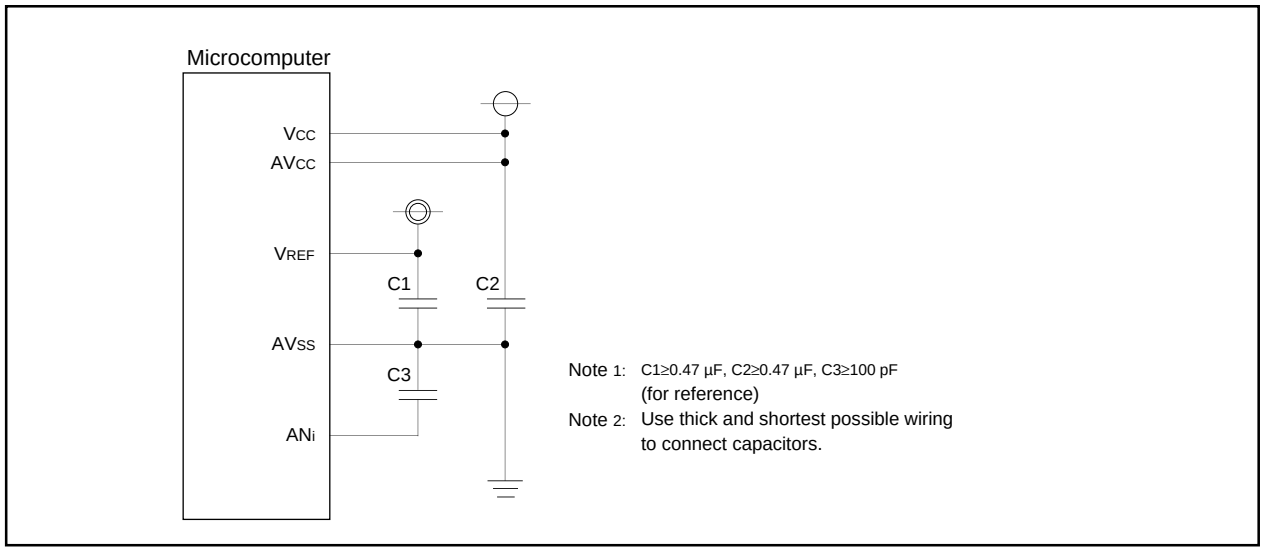


Figure 2.9.17. Use of capacitors to reduce noice

(3) Set the direction register of the following ports to input: the port corresponding to a pin to be used as an analog input pin and external trigger input pin (P93).

(4) Rewrite to analog input pin after changing A/D operation mode.

(5) When using the one-shot or single sweep mode

Confirm that A/D conversion is complete before reading the AD register.

(Note: When A/D conversion interrupt request bit is set, it shows that A/D conversion is completed.)

(6) When using the repeat mode or repeat sweep mode 0 or 1

Use the undivided main clock as the internal CPU clock.

(7) In using a key-input interrupt, none of the 8 pins (AN0 through AN7) can be used as an A/D conversion port (if the A/D input voltage goes to “L” level, a key-input interrupt occurs).

(8) Use ϕAD under 10 MHz. When XIN is over 10 MHz, divide it.

2.9.9 Method of A/D Conversion (10-bit mode)

(1) The A/D converter compares the reference voltage (Vref) generated internally based on the contents of the successive comparison register with the analog input voltage (V_{IN}) input from the analog input pin. The comparison result is stored in the successive comparison register and then V_{IN} is converted to digital value (successive comparison method). If a trigger occurs, the A/D converter carries out the following:

1. Fixes bit 9 of the successive comparison register.
   Compares Vref with V_{IN}: [In this instance, the contents of the successive comparison register are "1000000000_2" (default).]
   Bit 9 of the successive comparison register varies depending on the comparison result as follows.
   If Vref < V_{IN}, then "1" is assigned to bit 9.
   If Vref > V_{IN}, then "0" is assigned to bit 9.

2. Fixes bit 8 of the successive comparison register.
   Sets bit 8 of the successive comparison register to "1", then compares Vref with V_{IN}.
   Bit 8 of the successive comparison register varies depending on the comparison result as follows:
   If Vref < V_{IN}, then "1" is assigned to bit 8.
   If Vref > V_{IN}, then "0" is assigned to bit 8.

3. Fixes bit 7 through bit 0 of the successive comparison register.
   Carries out step 2 above on bit 7 through bit 0.
   After bit 0 is fixed, the contents of the successive comparison register (conversion result) are transmitted to AD register i.

Vref is generated based on the latest content of the successive comparison register. Table 2.9.8 shows the relationship of the successive comparison register contents and Vref. Table 2.9.9 shows how the successive comparison register and Vref vary while A/D conversion is in progress. Figure 2.9.18 shows theoretical A/D conversion characteristics.

Table 2.9.8. Relationship of the successive comparison register contents and Vref

Successive approximation register : n	Vref (V)
0	0
1 to1023	$\frac{V_{REF}}{1024} \times n - \frac{V_{REF}}{2048}$

Table 2.9.9. Variation of the successive comparison register and Vref while A/D conversion is in progress (10-bit mode)

	Successive approximation register	Vref change
A/D converter stopped	b9 b0 1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$ [V]
1st comparison	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{2048}$ [V]
2nd comparison	n9 1 0 0 0 0 0 0 0 0 n9: 1st comparison result	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{2048}$ [V] $\begin{pmatrix} n_9 = 1 & + \frac{V_{REF}}{4} \\ n_9 = 0 & - \frac{V_{REF}}{4} \end{pmatrix}$
3rd comparison	n9 n8 1 0 0 0 0 0 0 0 n8: 2nd comparison result	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{2048}$ [V] $\begin{pmatrix} n_8 = 1 & + \frac{V_{REF}}{8} \\ n_8 = 0 & - \frac{V_{REF}}{8} \end{pmatrix}$
10th comparison	n9 n8 n7 n6 n5 n4 n3 n2 n1 1	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} \pm \ldots\ldots \pm \frac{V_{REF}}{1024} - \frac{V_{REF}}{2048}$ [V]
Conversion complete	n9 n8 n7 n6 n5 n4 n3 n2 n1 n0 This data transfers to the bit 0 to bit 9 of AD register i.	

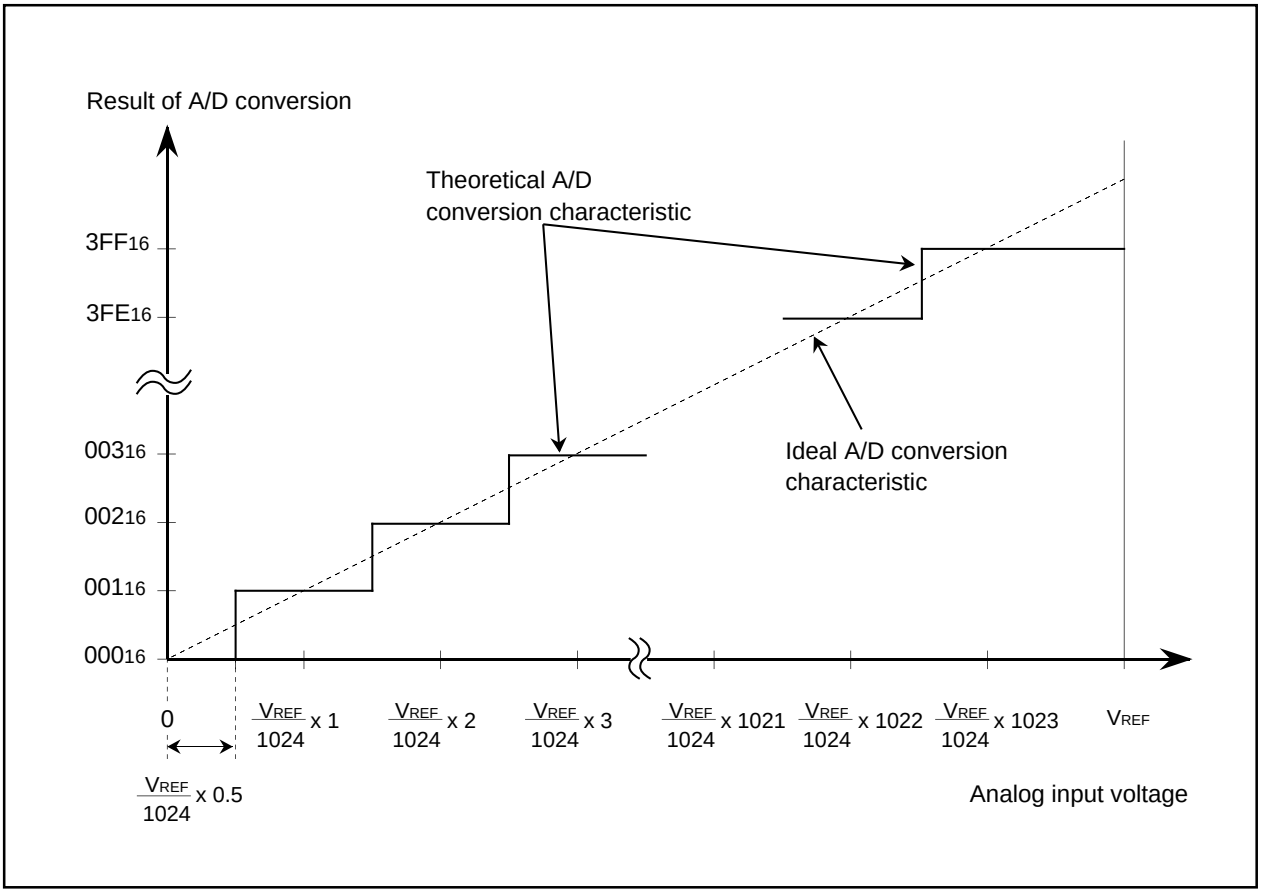


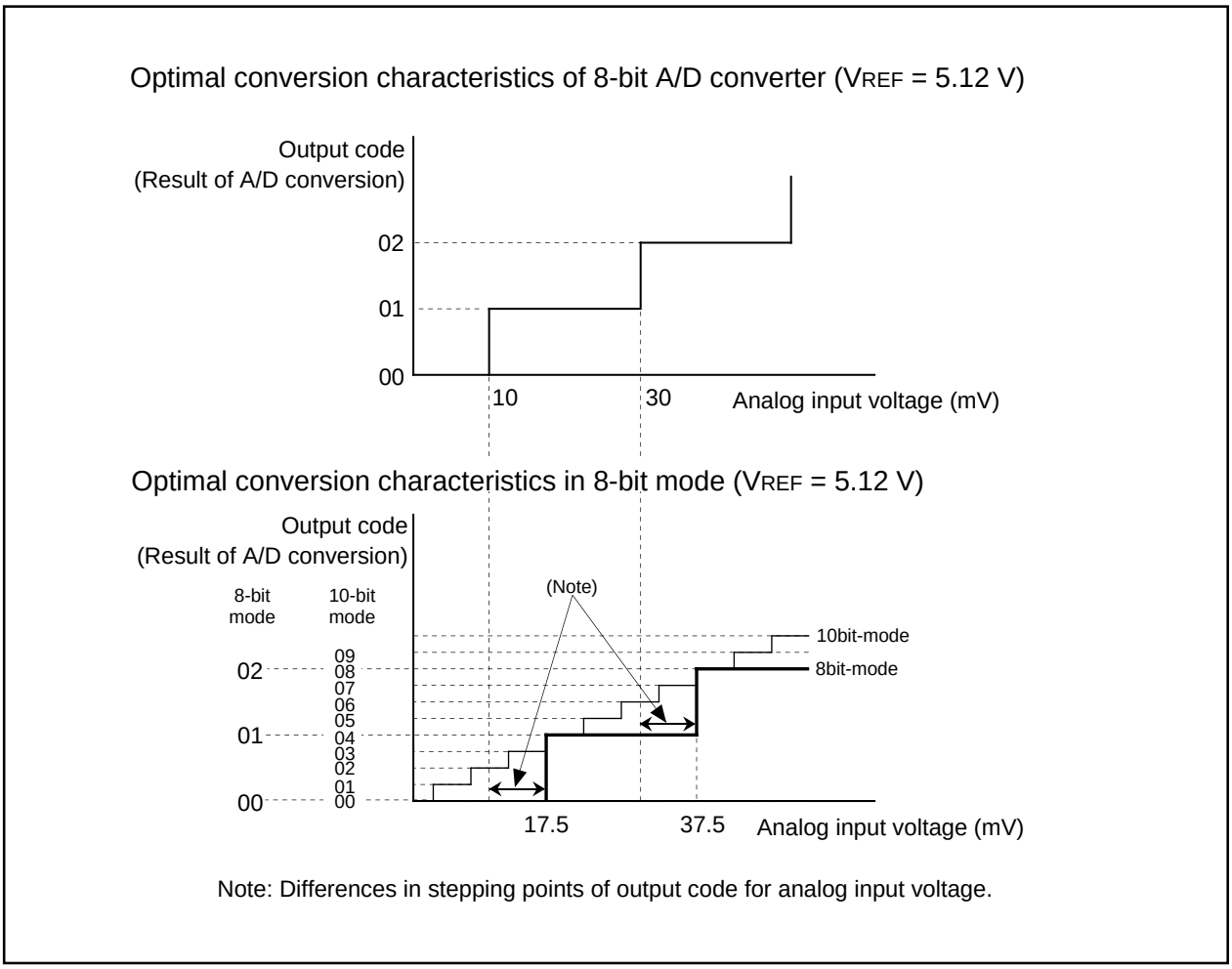
Figure 2.9.18. Theoretical A/D conversion characteristics (10-bit mode)

2.9.10 Method of A/D Conversion (8-bit mode)

(1) In 8-bit mode, 8 higher-order bits of the 10-bit successive comparison register becomes A/D conversion result. Hence, if compared to a result obtained by using an 8-bit A/D converter, the voltage compared is different by 3 VREF/2048 (see what are underscored in Table 2.9.10), and differences in stepping points of output codes occur as shown in Figure 2.9.19.

Table 2.9.10. The comparison voltage in 8-bit mode compared to 8-bit A/D converter

		8-bit mode	8-bit A/D converter
Comparison voltage Vref	n = 0	0	0
	n = 1 to 255	$\frac{VREF}{2^8} \times n - \frac{VREF}{2^{10}} \times 0.5$	$\frac{VREF}{2^8} \times n - \frac{VREF}{2^8} \times 0.5$



Note: Differences in stepping points of output code for analog input voltage.

Figure 2.9.19. The level conversion characteristics of 8-bit mode and 8-bit A/D converter

Table 2.9.11. Variation of the successive comparison register and Vref while A/D conversion is in progress (8-bit mode)

	Successive approximation register	Vref change
A/D converter stopped	b9 b0: 1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$ [V]
1st comparison	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{2048}$ [V]
2nd comparison	n9 1 0 0 0 0 0 0 0 0 (1st comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{2048}$ [V] ($n_9 = 1$: $+\frac{V_{REF}}{4}$; $n_9 = 0$: $-\frac{V_{REF}}{4}$)
3rd comparison	n9 n8 1 0 0 0 0 0 0 0 (2nd comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{2048}$ [V] ($n_8 = 1$: $+\frac{V_{REF}}{8}$; $n_8 = 0$: $-\frac{V_{REF}}{8}$)
8th comparison	n9 n8 n7 n6 n5 n4 n3 1 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} \pm \ldots\ldots \pm \frac{V_{REF}}{256} - \frac{V_{REF}}{2048}$ [V]
Conversion complete	n9 n8 n7 n6 n5 n4 n3 n2 0 0 — This data transfers to bit 0 to bit 7 of AD register i.	

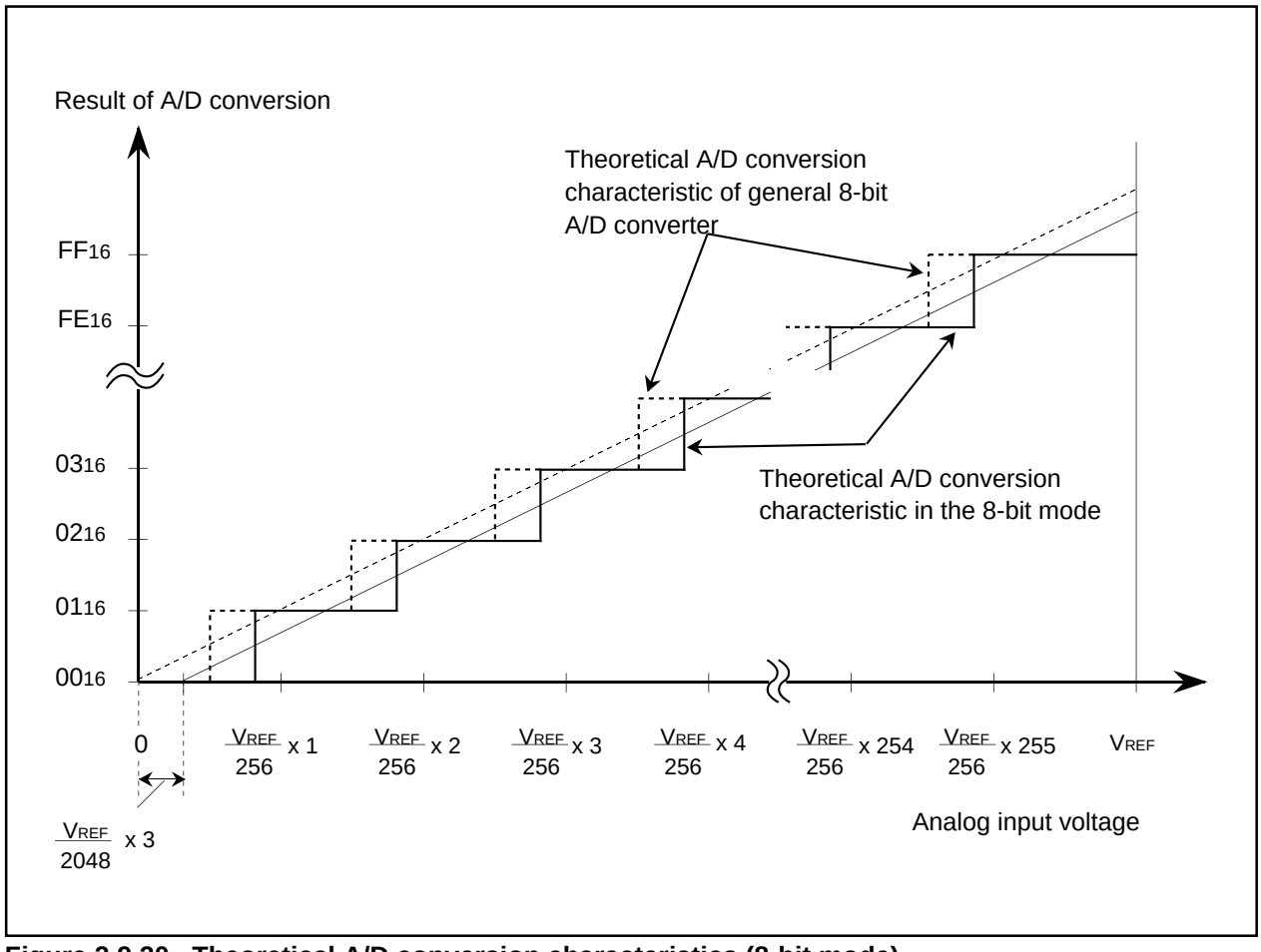


Figure 2.9.20. Theoretical A/D conversion characteristics (8-bit mode)

2.9.11 Absolute Accuracy and Differential Non-Linearity Error

- **Absolute accuracy**

  Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and actual A/D conversion result. When measuring absolute accuracy, the voltage at the middle point of the width of analog input voltage (1-LSB width), that can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as an analog input voltage. For example, if 10-bit resolution is used and if V_{REF} (reference voltage) = 5.12 V, then 1-LSB width becomes 5 mV, and 0 mV, 5 mV, 10 mV, 15 mV, 20 mV, ····are used as analog input voltages. If analog input voltage is 25 mV, "absolute accuracy = ± 3LSB" refers to the fact that actual A/D conversion falls on a range from "002_{16}" to "008_{16}" though an output code, "005_{16}", can be expected from the theoretical A/D conversion characteristics. Zero error and full-scale error are included in absolute accuracy.

  Also, all the output codes for analog input voltage between V_{REF} and AVcc becomes "$3FF_{16}$".

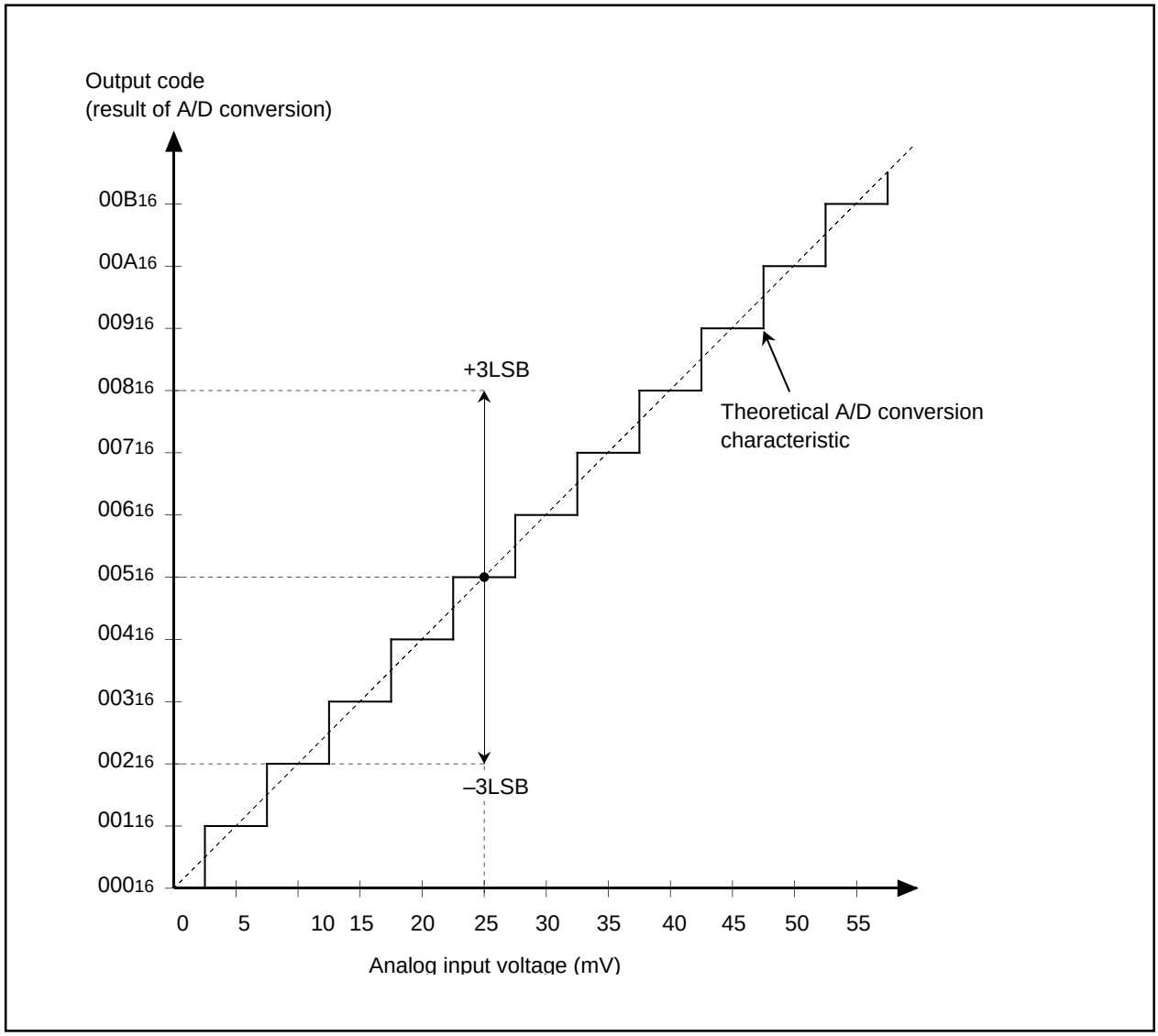


Figure 2.9.21. Absolute accuracy (10-bit resolution)

• **Differential non-linearity error**

Differential non-linearity error refers to the difference between 1-LSB width based on the theoretical A/D conversion characteristics (an analog input width that can meet the expectation of outputting an equal code) and an actually measured 1-LSB width (analog input voltage width that outputs an equal code). If 10-bit resolution is used and if V_{REF} (reference voltage) = 5.12 V, “differential non-linearity error = ± 1LSB” refers to the fact that 1-LSB width actually measured falls on a range from 0 mV to 10 mV though 1-LSB width based on the theoretical A/D conversion characteristics is 5 mV.

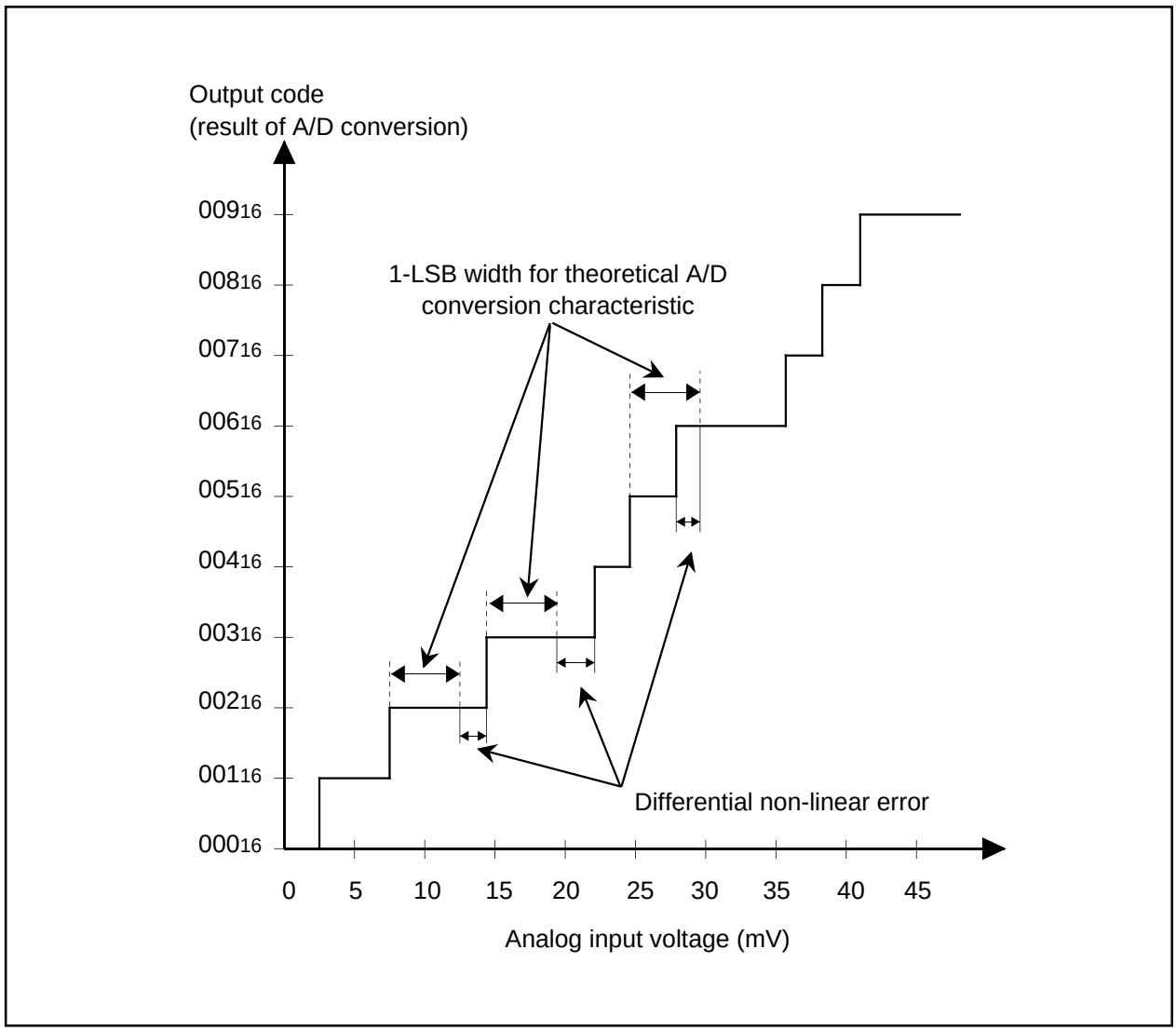
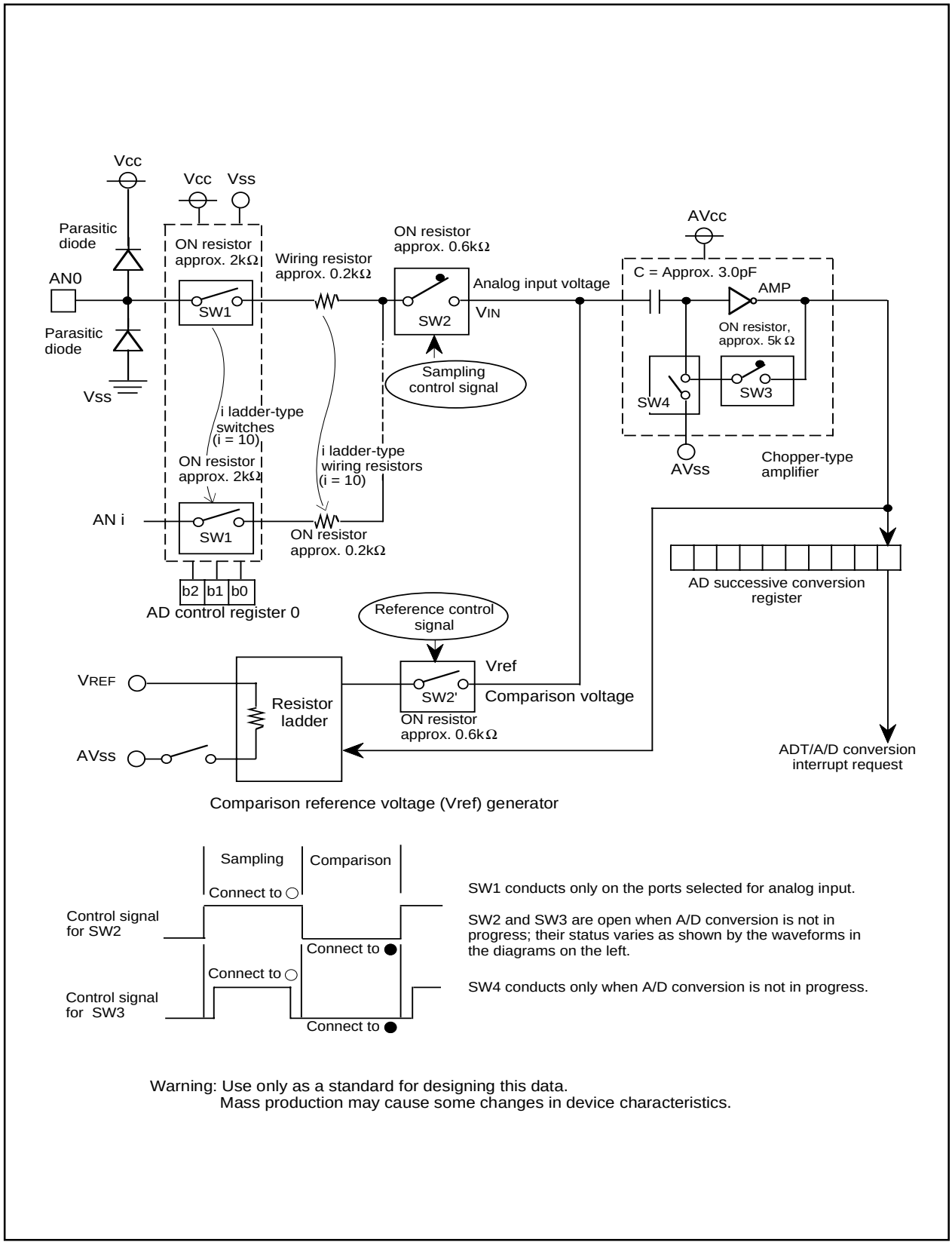


Figure 2.9.22. Differential non-linearity error (10-bit resolution)

2.9.12 Internal Equivalent Circuit of Analog Input

Figure 2.9.23 shows the internal equivalent circuit of analog input.



Warning: Use only as a standard for designing this data.
Mass production may cause some changes in device characteristics.

Figure 2.9.23. Internal equivalent circuit to analog input

2.9.13 Sensor's Output Impedance under A/D Conversion (reference value)

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 2.9.23 has to be completed within a specified period of time. With T as the specified time, time T is the time that switches SW2 and SW3 are connected to O in Figure 2.9.23. Let output impedance of sensor equivalent circuit be R0, microcomputer's internal resistance be R, precision (error) of the A/D converter be X, and the A/D converter's resolution be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

Vc is generally $V_C = V_{IN}\{1 - e^{-\frac{t}{C(R0 + R)}}\}$

And when t = T, $V_C = V_{IN} - \frac{X}{Y}V_{IN} = V_{IN}(1 - \frac{X}{Y})$

$$e^{-\frac{T}{C(R0 + R)}} = \frac{X}{Y}$$

$$-\frac{T}{C(R0 + R)} = \ln\frac{X}{Y}$$

Hence, $R0 = -\frac{T}{C \cdot \ln\frac{X}{Y}} - R$

With the model shown in Figure 2.9.24 as an example, when the difference between V_{IN} and V_C becomes 0.1LSB, we find impedance R0 when voltage between pins V_C changes from 0 to V_{IN}-(0.1/1024) V_{IN} in time T. (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is held to 0.1LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB. When $f(X_{IN})$ = 10 MHz, T = 0.3 us in the A/D conversion mode with sample & hold. Output impedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.3 µs, R = 7.8 kΩ, C = 3 pF, X = 0.1, and Y = 1024 . Hence,

$$R0 = -\frac{0.3 \times 10^{-6}}{3.0 \times 10^{-12} \cdot \ln\frac{0.1}{1024}} - 7.8 \times 10^{3} \fallingdotseq 3.0 \times 10^{3}$$

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A/D converter turns out to be approximately 3.0 kΩ. Tables 2.9.12 and 2.9.13 show output impedance values based on the LSB values.

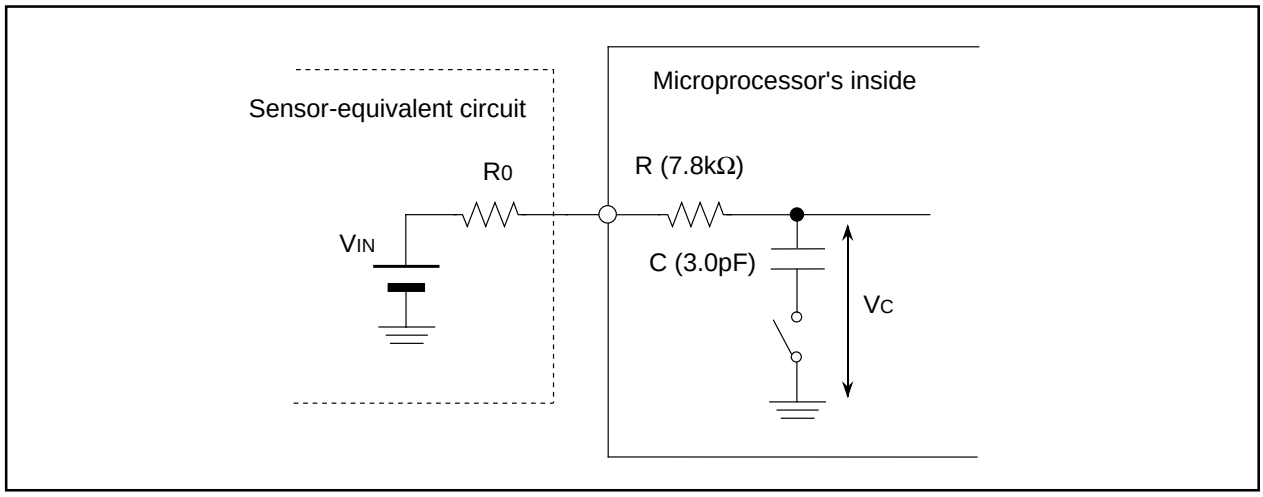


Figure 2.9.24 A circuit equivalent to the A/D conversion terminal

Table 2.9.12. Relation between output impedance and precision (error) of A/D converter (10-bit mode) Reference value

f(Xin) (MHz)	Cycle (µs)	Sampling time (µs)	R (k)	C (pF)	Resolution (LSB)	R0 (k)
10	0.1	0.3 (3 x cycle, Sample & hold bit is enabled)	7.8	3.0	0.1	3.0
					0.3	4.5
					0.5	5.3
					0.7	5.9
					0.9	6.4
					1.1	6.8
					1.3	7.2
					1.5	7.5
					1.7	7.8
					1.9	8.1
10	0.1	0.2 (2 x cycle, Sample & hold bit is disabled)	7.8	3.0	0.3	0.4
					0.5	0.9
					0.7	1.3
					0.9	1.7
					1.1	2.0
					1.3	2.2
					1.5	2.4
					1.7	2.6
					1.9	2.8

Table 2.9.13. Relation between output impedance and precision (error) of A/D converter (8-bit mode) Reference value

f(Xin) (MHz)	Cycle (µs)	Sampling time (µs)	R (k)	C (pF)	Resolution (LSB)	R0 (k)
10	0.1	0.3 (3 x cycle, Sample & hold bit is enabled)	7.8	3.0	0.1	4.9
					0.3	7.0
					0.5	8.2
					0.7	9.1
					0.9	9.9
					1.1	10.5
					1.3	11.1
					1.5	11.7
					1.7	12.1
					1.9	12.6
10	0.1	0.2 (2 x cycle, Sample & hold bit is disabled)	7.8	3.0	0.1	0.7
					0.3	2.1
					0.5	2.9
					0.7	3.5
					0.9	4.0
					1.1	4.4
					1.3	4.8
					1.5	5.2
					1.7	5.5
					1.9	5.8

2.10 DMAC Usage

2.10.1 Overview of the DMAC usage

DMAC transfers one data item held in the source address to the destination address every time a transfer request is generated. The following is an overview of the DMAC usage.

(1) Source address and destination address

Both the register which indicates a source and the register which indicates a destination comprise of 24 bits, so that each can cover a 1M bytes space. After transfer of one bit of data is completed, the address in either the source register or the destination register can be incremented. However, both registers cannot be incremented. The links between the source and destination are as follows:

(a) A fixed address from an arbitrary 1M bytes space

(b) An arbitrary 1M bytes space from a fixed address

(c) A fixed address from another fixed address

([0020_{16} to $003F_{16}$ and 0180_{16} to $019F_{16}$] cannot be accessed)

(2) The number of bits of data transferred

The number of bit of data indicated by the transfer counter is transferred. If a 16-bit transfer is selected, up to 128K bytes can be transferred. If an 8-bit transfer is selected, up to 64K bytes can be transferred. The transfer counter is decremented each time one bit of data is transferred, and a DMA interrupt request occurs when the transfer counter underflows.

(3) DMA transfer factor

The DMA transfer factor can be selected from the following 31 factors: falling edge/two edges of $\overline{INT0}$/$\overline{INT1}$/$\overline{INT2}$ pin, timer A0 interrupt request through timer A4 interrupt request, UART0 transmission/NACK/SS interface 0 transmission interrupt request, UART0 reception/ACK/SS interface 0 reception interrupt request, UART1 transmission/NACK/SS interface 1 transmission interrupt request, UART1 reception/ACK/SS interface 1 reception interrupt request, UART2 transmission/NACK interrupt request, UART2 reception/ACK interrupt request, UART3 transmission/NACK interrupt request, UART3 reception/ACK interrupt request, USB0/USB1/USB2/USB3 function interrupt request, A/D conversion interrupt request, software trigger, and DMA trigger.

Software trigger is always enabled. When software trigger is selected, DMA transfer is generated by writing “1” to software DMA interrupt request bit. When other factor is selected, DMA transfer is generated by generating corresponding interrupt request.

(4) Channel priority

High to low priority: DMA0, DMA1, DMA2, DMA3

(5) Writing to a register

When writing to the source register or the destination register with DMA enabled, the content of the register with a fixed address will change at the time of writing. Therefore, the user should not write to a register with a fixed address when the DMA enable bit is set to “1”. The contents of the register with ‘forward direction’ selected, and the transfer counter, are changed when reloaded. A reload occurs either when the transfer counter underflows, or when the DMA enable bit is re-enabled, after having been disabled.

The reload register can be written to, as in normal conditions.

(6) Reading to a register

The reload register can be read to, as in normal conditions.

(7) Switching function

(a) Switching between one-shot transfer and repeated transfer

'One-shot transfer' refers to a mode in which DMA is disabled after the transfer counter underflows. 'Repeated transfer' refers to a mode in which a reload is carried out after the transfer counter underflows. The reload is carried out for the transfer counter and on the address pointer subjected to forward direction.

The following examples are described in section 2.10.2 and 2.10.3.

- A fixed address from an arbitrary 1M byte space, one-shot transfer
- An arbitrary 1M byte space from a fixed address, repeated transfer

(8) Registers related to DMAC

Figure 2.10.1 shows the memory map of DMAC-related registers, and Figures 2.10.2 and 2.10.4 show DMAC-related registers.

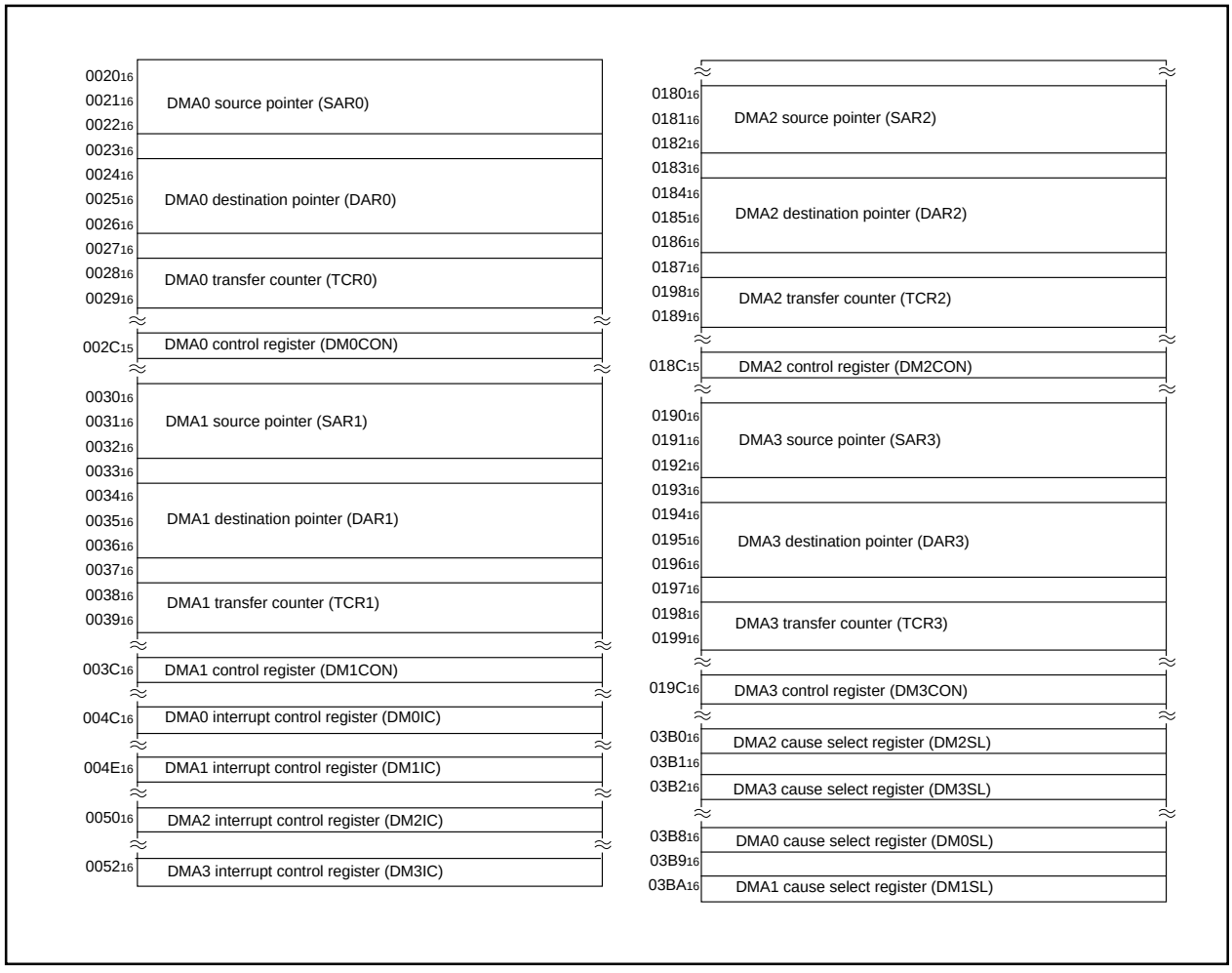
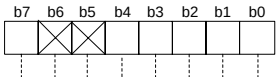

0020₁₆ 0021₁₆ 0022₁₆ 0023₁₆ DMA0 source pointer (SAR0)
0024₁₆ 0025₁₆ 0026₁₆ 0027₁₆ DMA0 destination pointer (DAR0)
0028₁₆ 0029₁₆ DMA0 transfer counter (TCR0)
002C₁₆ DMA0 control register (DM0CON)
0030₁₆ 0031₁₆ 0032₁₆ 0033₁₆ DMA1 source pointer (SAR1)
0034₁₆ 0035₁₆ 0036₁₆ 0037₁₆ DMA1 destination pointer (DAR1)
0038₁₆ 0039₁₆ DMA1 transfer counter (TCR1)
003C₁₆ DMA1 control register (DM1CON)
004C₁₆ DMA0 interrupt control register (DM0IC)
004E₁₆ DMA1 interrupt control register (DM1IC)
0050₁₆ DMA2 interrupt control register (DM2IC)
0052₁₆ DMA3 interrupt control register (DM3IC)
0180₁₆ 0181₁₆ 0182₁₆ 0183₁₆ DMA2 source pointer (SAR2)
0184₁₆ 0185₁₆ 0186₁₆ 0187₁₆ DMA2 destination pointer (DAR2)
0198₁₆ 0189₁₆ DMA2 transfer counter (TCR2)
018C₁₆ DMA2 control register (DM2CON)
0190₁₆ 0191₁₆ 0192₁₆ 0193₁₆ DMA3 source pointer (SAR3)
0194₁₆ 0195₁₆ 0196₁₆ 0197₁₆ DMA3 destination pointer (DAR3)
0198₁₆ 0199₁₆ DMA3 transfer counter (TCR3)
019C₁₆ DMA3 control register (DM3CON)
03B0₁₆ DMA2 cause select register (DM2SL)
03B1₁₆ 03B2₁₆ DMA3 cause select register (DM3SL)
03B8₁₆ DMA0 cause select register (DM0SL)
03B9₁₆ 03BA₁₆ DMA1 cause select register (DM1SL)

Figure 2.10.1. Memory map of DMAC-related registers

DMA0 request cause select register (Note 1)



Symbol: DM0SL | Address: 03B8$_{16}$ | When reset: 00$_{16}$

Bit Symbol	Bit Name	Function (Note 2)	R	W
DSEL0	DMA request cause select bits	b4 b3 b2 b1 b0 0 0 0 0 0 : DMA Disabled 0 0 0 0 1 : INT0 (falling edge) 0 0 0 1 0 : INT0 (two edges) 0 0 0 1 1 : USB0 0 0 1 0 0 : Timer A0 0 0 1 0 1 : Timer A1 0 0 1 1 0 : Timer A2 0 0 1 1 1 : Timer A3 0 1 0 0 0 : Timer A4 0 1 0 0 1 : UART0 receive/ACK/SSI0 receive 0 1 0 1 0 : UART1 receive/ACK/SSI1 receive 0 1 0 1 1 : UART2 receive/ACK 0 1 1 0 0 : UART3 receive/ACK 0 1 1 0 1 : UART0 transmit/NACK/SSI0 transmit 0 1 1 1 0 : UART1 transmit/NACK/SSI1 transmit 0 1 1 1 1 : UART2 transmit/NACK 1 0 0 0 0 : UART3 transmit/NACK 1 0 0 0 1 : A/D 1 0 0 1 0 : Disabled (Note 3) 1 0 0 1 1 : DMA1 1 0 1 0 0 : DMA2 1 0 1 0 1 : DMA3 1 0 1 1 0 : Disabled (Note 3) 1 0 1 1 1 : Disabled (Note 3) 1 1 x x x : Disabled (Note 3)	O	O
DSEL1			O	O
DSEL2			O	O
DSEL3			O	O
DSEL4			O	O
Nothing is assigned. Write “0” when writing to these bits. The value is “0” when read.			–	–
DSR	Software DMA request bit	Software trigger is always enabled Write “1” to trigger DSR bit.	O	O

Note 1: Software is always enabled.
Note 2: SSI=Serial sound interface
Note 3: This value should not be set.

DMA1 request cause select register (Note 1)

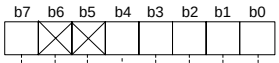
Symbol: DM1SL | Address: 03BA$_{16}$ | When reset: 00$_{16}$

Bit Symbol	Bit Name	Function (Note 2)	R	W
DSEL0	DMA request cause select bits	b4 b3 b2 b1 b0 0 0 0 0 0 : DMA Disabled 0 0 0 0 1 : INT1 (falling edge) 0 0 0 1 0 : INT1 (two edges) 0 0 0 1 1 : USB1 0 0 1 0 0 : Timer A0 0 0 1 0 1 : Timer A1 0 0 1 1 0 : Timer A2 0 0 1 1 1 : Timer A3 0 1 0 0 0 : Timer A4 0 1 0 0 1 : UART0 receive/ACK/SSI0 receive 0 1 0 1 0 : UART1 receive/ACK/SSI1 receive 0 1 0 1 1 : UART2 receive/ACK 0 1 1 0 0 : UART3 receive/ACK 0 1 1 0 1 : UART0 transmit/NACK/SSI0 transmit 0 1 1 1 0 : UART1 transmit/NACK/SSI1 transmit 0 1 1 1 1 : UART2 transmit/NACK 1 0 0 0 0 : UART3 transmit/NACK 1 0 0 0 1 : A/D 1 0 0 1 0 : DMA0 1 0 0 1 1 : Disabled (Note 3) 1 0 1 0 0 : DMA2 1 0 1 0 1 : DMA3 1 0 1 1 0 : Disabled (Note 3) 1 0 1 1 1 : Disabled (Note 3) 1 1 x x x : Disabled (Note 3)	O	O
DSEL1			O	O
DSEL2			O	O
DSEL3			O	O
DSEL4			O	O
Nothing is assigned. Write “0” when writing to these bits. The value is “0” when read.			–	–
DSR	Software DMA request bit	Software trigger is always enabled Write “1” to trigger DSR bit.	O	O

Note 1: Software is always enabled.
Note 2: SSI=Serial sound interface
Note 3: This value should not be set.

Figure 2.10.2. DMAC-related registers (1)

DMA2 request cause select register (Note 1)

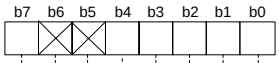


Symbol: DM2SL | Address: 03B016 | When reset: 0016

Bit Symbol	Bit Name	Function (Note 2)	R	W
DSEL0	DMA request cause select bits	b4 b3 b2 b1 b0 0 0 0 0 0 : DMA Disabled 0 0 0 0 1 : INT2 (falling edge) 0 0 0 1 0 : INT2 (two edges) 0 0 0 1 1 : USB2 0 0 1 0 0 : Timer A0 0 0 1 0 1 : Timer A1 0 0 1 1 0 : Timer A2 0 0 1 1 1 : Timer A3 0 1 0 0 0 : Timer A4 0 1 0 0 1 : UART0 receive/ACK/SSI0 receive 0 1 0 1 0 : UART1 receive/ACK/SSI1 receive 0 1 0 1 1 : UART2 receive/ACK 0 1 1 0 0 : UART3 receive/ACK 0 1 1 0 1 : UART0 transmit/NACK/SSI0 transmit 0 1 1 1 0 : UART1 transmit/NACK/SSI1 transmit 0 1 1 1 1 : UART2 transmit/NACK 1 0 0 0 0 : UART3 transmit/NACK 1 0 0 0 1 : A/D 1 0 0 1 0 : DMA0 1 0 0 1 1 : DMA1 1 0 1 0 0 : Disabled (Note 3) 1 0 1 0 1 : DMA3 1 0 1 1 0 : Disabled (Note 3) 1 0 1 1 1 : Disabled (Note 3) 1 1 x x x : Disabled (Note 3)	O	O
DSEL1				
DSEL2				
DSEL3				
DSEL4				
Nothing is assigned. Write “0” when writing to these bits. The value is “0” when read.			–	–
DSR	Software DMA request bit	Software trigger is always enabled Write “1” to trigger DSR bit.	O	O

Note 1: Software is always enabled.
Note 2: SSI=Serial sound interface
Note 3: This value should not be set.

DMA3 request cause select register (Note 1)



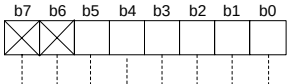
Symbol: DM3SL | Address: 03B216 | When reset: 0016

Bit Symbol	Bit Name	Function (Note 2)	R	W
DSEL0	DMA request cause select bits	b4 b3 b2 b1 b0 0 0 0 0 0 : DMA Disabled 0 0 0 0 1 : INT0 (falling edge) 0 0 0 1 0 : INT0 (two edges) 0 0 0 1 1 : USB3 0 0 1 0 0 : Timer A0 0 0 1 0 1 : Timer A1 0 0 1 1 0 : Timer A2 0 0 1 1 1 : Timer A3 0 1 0 0 0 : Timer A4 0 1 0 0 1 : UART0 receive/ACK/SSI0 receive 0 1 0 1 0 : UART1 receive/ACK/SSI1 receive 0 1 0 1 1 : UART2 receive/ACK 0 1 1 0 0 : UART3 receive/ACK 0 1 1 0 1 : UART0 transmit/NACK/SSI0 transmit 0 1 1 1 0 : UART1 transmit/NACK/SSI1 transmit 0 1 1 1 1 : UART2 transmit/NACK 1 0 0 0 0 : UART3 transmit/NACK 1 0 0 0 1 : A/D 1 0 0 1 0 : DMA0 1 0 0 1 1 : DMA1 1 0 1 0 0 : DMA2 1 0 1 0 1 : Disabled (Note 3) 1 0 1 1 0 : Disabled (Note 3) 1 0 1 1 1 : Disabled (Note 3) 1 1 x x x : Disabled (Note 3)	O	O
DSEL1				
DSEL2				
DSEL3				
DSEL4				
Nothing is assigned. Write “0” when writing to these bits. The value is “0” when read.			–	–
DSR	Software DMA request bit	Software trigger is always enabled Write “1” to trigger DSR bit.	O	O

Note 1: Software is always enabled.
Note 2: SSI=Serial sound interface
Note 3: This value should not be set.

Figure 2.10.3. DMAC-related registers (2)

DMAi control register

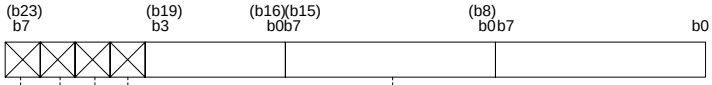


Symbol	Address	When reset
DMiCON (i=0-3)	002C16, 003C16, 018C16, 019C16	00000X002

Bit Symbol	Bit Name	Function	R	W
DMBIT	Transfer unit select bit	0 : 16 bits 1 : 8 bits	O	O
DMASL	Repeat transfer mode select bit	0 : Single transfer 1 : Repeat transfer	O	O
DMAS	DMA request bit (Note 1)	0 : DMA not requested 1 : DMA requested	O	O (Note 2)
DMAE	DMA enable bit	0 : Disabled 1 : Enabled	O	O
DSD	Source address direction select bit (Note 3)	0 : Fixed 1 : Forward	O	O
DAD	Destination address direction select bit (Note 3)	0 : Fixed 1 : Forward	O	O
Nothing is assigned. Write “0” when writing to these bits. The value is “0” when read.			–	–

Note 1: DMA request can be cleared by resetting the bit.
Note 2: This bit can only be set to “0”.
Note 3: Source address direction select bit and destination address direction select bit cannot be set to “1” simultaneously.

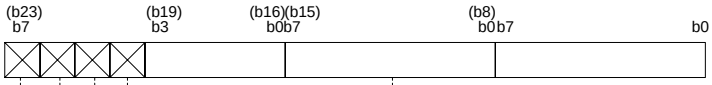
DMAi source pointer (i=0-3)



Symbol	Address	When reset
SAR0	002216 to 002016	Indeterminate
SAR1	003216 to 003016	Indeterminate
SAR2	018216 to 018016	Indeterminate
SAR3	019216 to 019016	Indeterminate

Function	Transfer address specification	R	W
Source pointer stores the source address	000016 to FFFFF16	O	O
Nothing is assigned. Write “0” when writing to these bits. The value is “0” when read.		–	–

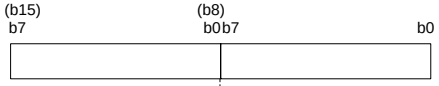
DMAi destination pointer (i=0-3)



Symbol	Address	When reset
DAR0	002616 to 002416	Indeterminate
DAR1	003616 to 003416	Indeterminate
DAR2	018616 to 018416	Indeterminate
DAR3	019616 to 019416	Indeterminate

Function	Transfer address specification	R	W
Destination pointer stores the destination address	000016 to FFFFF16	O	O
Nothing is assigned. Write “0” when writing to these bits. The value is “0” when read.		–	–

DMAi transfer counter (i=0-3)



Symbol	Address	When reset
TCR0	002916 to 002816	Indeterminate
TCR1	003916 to 003816	Indeterminate
TCR2	018916 to 018816	Indeterminate
TCR3	019816 to 019816	Indeterminate

Function	Transfer count specification	R	W
Transfer counter Set a value one less than the transfer count	000016 to FFFFF16	O	O

Figure 2.10.4. DMAC-related registers (3)

2.10.2 Operation of DMAC (one-shot transfer mode)

In one-shot transfer mode, choose functions from the items shown in Table 2.10.1. Operations of the circled items are described below. Figure 2.10.5 shows an example of operation and Figure 2.10.6 shows the set-up procedure.

Table 2.10.1. Choosed functions

Item	Set-up	
Transfer space	**O**	Fixed address from an arbitrary 1 M bytes space
		Arbitrary 1 M bytes space from a fixed address
		Fixed address from fixed address
Unit of transfer	**O**	8 bits
		16 bits

Operation

(1) When software trigger is selected, setting software DMA request bit to “1” generates a DMA transfer request signal.

(2) If DMAC is active, data transfer starts, and the contents of the address indicated by the DMAi forward-direction address pointer are transferred to the address indicated by the DMAi destination pointer. When data transfer starts directly after DMAC becomes active, the value of the DMAi transfer counter reload register is reloaded to the DMAi transfer counter, and the value of the DMAi source pointer is reloaded by the DMAi forward-direction address pointer. Each time a DMA transfer request signal is generated, 1 byte of data is transferred. The DMAi transfer counter is down counted, and the DMAi forward-direction address pointer is up counted.

(3) If the DMA transfer counter underflows, the DMA enable bit changes to “0” and DMA transfer is completed. The DMA interrupt request bit changes to “1” simultaneously.

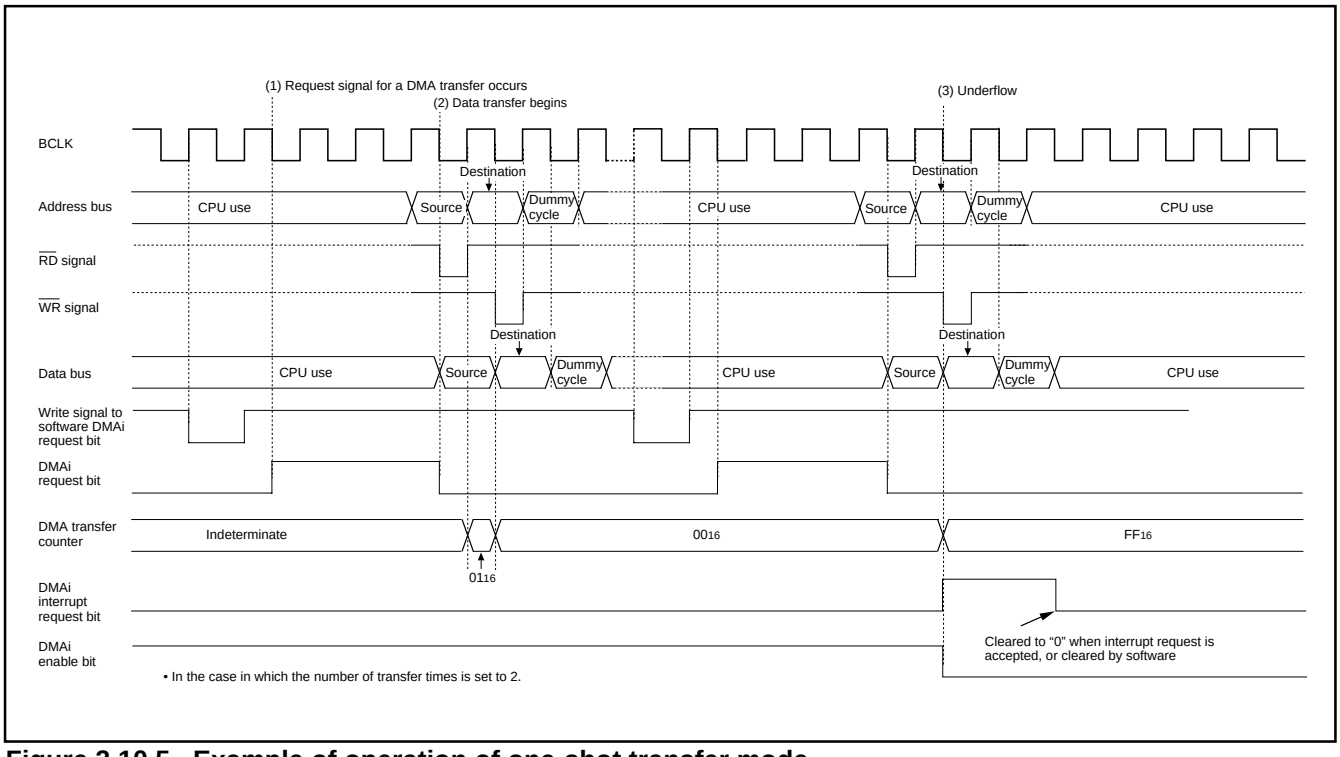


Figure 2.10.5. Example of operation of one-shot transfer mode

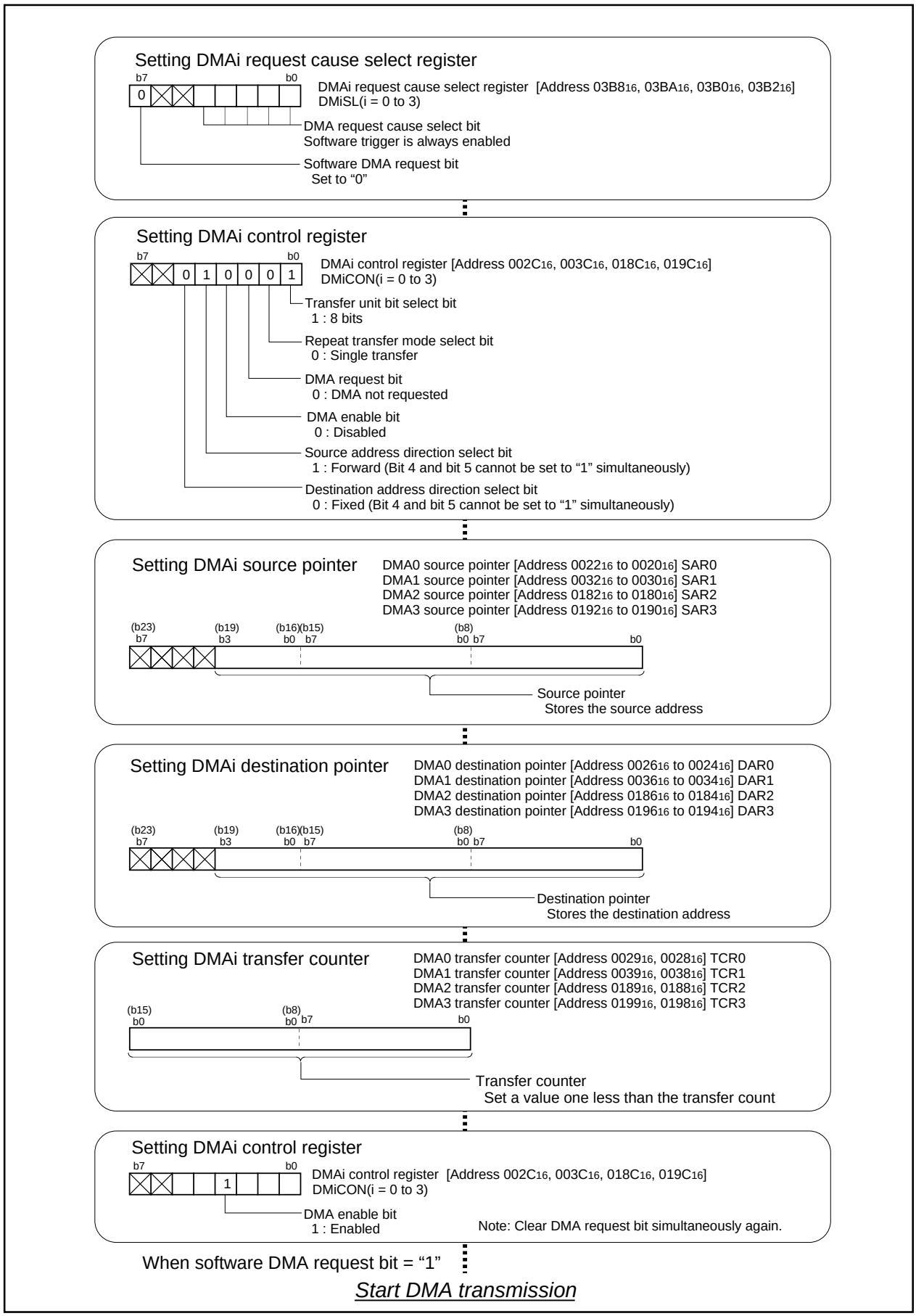

Setting DMAi request cause select register
b7 b0
0
DMAi request cause select register [Address 03B816, 03BA16, 03B016, 03B216]
DMiSL(i = 0 to 3)
DMA request cause select bit
Software trigger is always enabled
Software DMA request bit
Set to "0"
Setting DMAi control register
b7 b0
0 1 0 0 0 1
DMAi control register [Address 002C16, 003C16, 018C16, 019C16]
DMiCON(i = 0 to 3)
Transfer unit bit select bit
1 : 8 bits
Repeat transfer mode select bit
0 : Single transfer
DMA request bit
0 : DMA not requested
DMA enable bit
0 : Disabled
Source address direction select bit
1 : Forward (Bit 4 and bit 5 cannot be set to "1" simultaneously)
Destination address direction select bit
0 : Fixed (Bit 4 and bit 5 cannot be set to "1" simultaneously)
Setting DMAi source pointer
DMA0 source pointer [Address 002216 to 002016] SAR0
DMA1 source pointer [Address 003216 to 003016] SAR1
DMA2 source pointer [Address 018216 to 018016] SAR2
DMA3 source pointer [Address 019216 to 019016] SAR3
(b23) (b19) (b16)(b15) (b8)
b7 b3 b0 b7 b0 b7 b0
Source pointer
Stores the source address
Setting DMAi destination pointer
DMA0 destination pointer [Address 002616 to 002416] DAR0
DMA1 destination pointer [Address 003616 to 003416] DAR1
DMA2 destination pointer [Address 018616 to 018416] DAR2
DMA3 destination pointer [Address 019616 to 019416] DAR3
(b23) (b19) (b16)(b15) (b8)
b7 b3 b0 b7 b0 b7 b0
Destination pointer
Stores the destination address
Setting DMAi transfer counter
DMA0 transfer counter [Address 002916, 002816] TCR0
DMA1 transfer counter [Address 003916, 003816] TCR1
DMA2 transfer counter [Address 018916, 018816] TCR2
DMA3 transfer counter [Address 019916, 019816] TCR3
(b15) (b8)
b0 b0 b7 b0
Transfer counter
Set a value one less than the transfer count
Setting DMAi control register
b7 b0
1
DMAi control register [Address 002C16, 003C16, 018C16, 019C16]
DMiCON(i = 0 to 3)
DMA enable bit
1 : Enabled
Note: Clear DMA request bit simultaneously again.
When software DMA request bit = "1"
Start DMA transmission

Figure 2.10.6. Set-up procedure of one-shot transfer mode

2.10.3 Operation of DMAC (repeated transfer mode)

In repeat transfer mode, choose functions from the items shown in Table 2.10.2. Operations of the circled items are described below. Figure 2.10.7 shows an example of operation and Figure 2.10.8 shows the set-up procedure.

Table 2.10.2. Choosed functions

Item	Set-up	
Transfer space		Fixed address from an arbitrary 1 M bytes space
	O	Arbitrary 1 M bytes space from a fixed address
		Fixed address from fixed address
Unit of transfer		8 bits
	O	16 bits

Operation (1) When software trigger is selected, setting software DMA request bit to “1” generates a DMA transfer request signal.

(2) If DMAC is active, data transfer starts, and the contents of the address indicated by the DMAi forward-direction address pointer are transferred to the address indicated by the DMAi destination pointer. When data transfer starts directly after DMAC becomes active, the value of the DMAi transfer counter reload register is reloaded to the DMAi transfer counter, and the value of the DMAi source pointer is reloaded by the DMAi forward-direction address pointer. Each time a DMA transfer request signal is generated, 2 byte of data is transferred. The DMAi transfer counter is down counted, and the DMAi forward-direction address pointer is up counted.

(3) Though DMAi transfer counter is underflowed, DMA enable bit is still “1”. The DMA interrupt request bit changes to “1” simultaneously.

(4) After DMAi transfer counter is underflowed, when the next DMA request is generated, DMA transfer is repeated from (1).

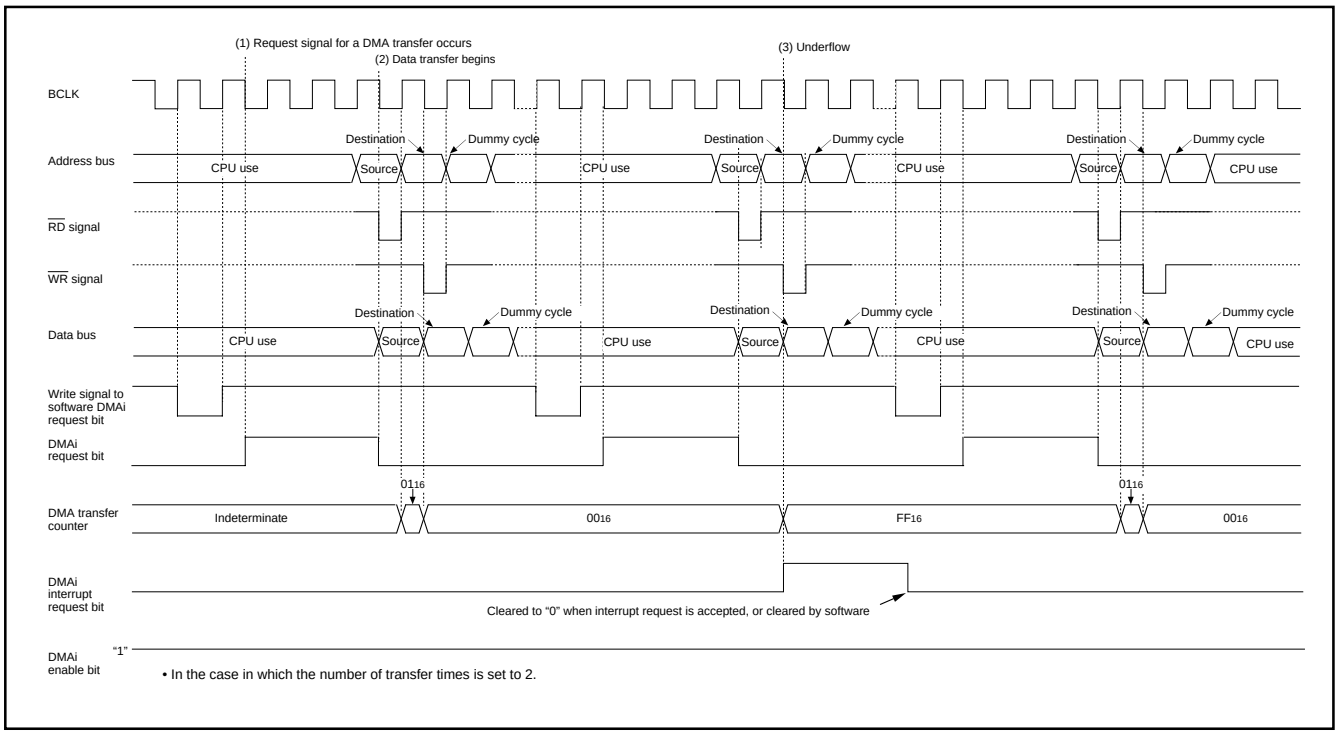


Figure 2.10.7. Example of operation of repeated transfer mode

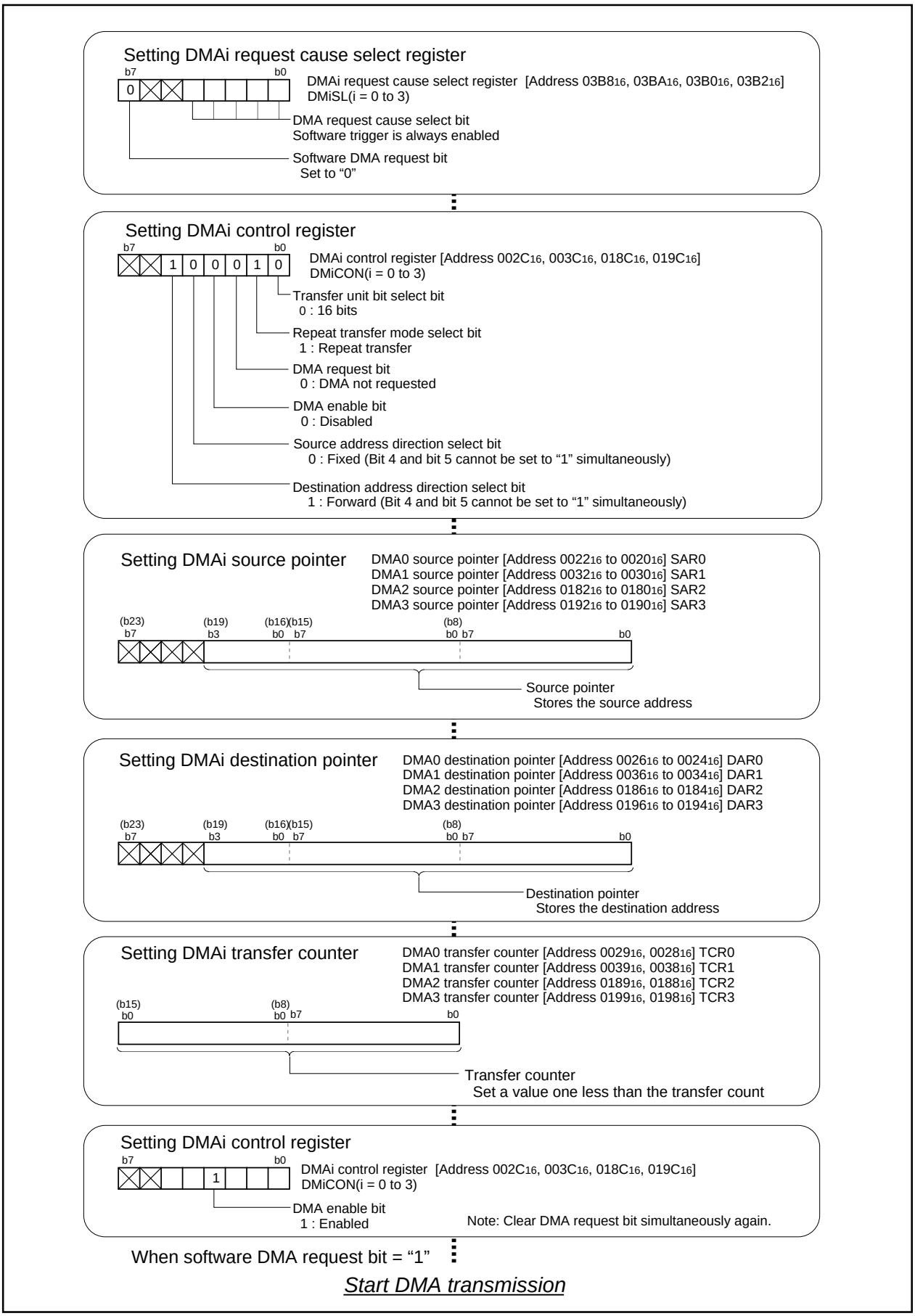

Setting DMAi request cause select register
b7 b0
0
DMAi request cause select register [Address 03B816, 03BA16, 03B016, 03B216]
DMiSL(i = 0 to 3)
DMA request cause select bit
Software trigger is always enabled
Software DMA request bit
Set to "0"
Setting DMAi control register
b7 b0
1 0 0 0 1 0
DMAi control register [Address 002C16, 003C16, 018C16, 019C16]
DMiCON(i = 0 to 3)
Transfer unit bit select bit
0 : 16 bits
Repeat transfer mode select bit
1 : Repeat transfer
DMA request bit
0 : DMA not requested
DMA enable bit
0 : Disabled
Source address direction select bit
0 : Fixed (Bit 4 and bit 5 cannot be set to "1" simultaneously)
Destination address direction select bit
1 : Forward (Bit 4 and bit 5 cannot be set to "1" simultaneously)
Setting DMAi source pointer
DMA0 source pointer [Address 002216 to 002016] SAR0
DMA1 source pointer [Address 003216 to 003016] SAR1
DMA2 source pointer [Address 018216 to 018016] SAR2
DMA3 source pointer [Address 019216 to 019016] SAR3
(b23) (b19) (b16)(b15) (b8)
b7 b3 b0 b7 b0 b7 b0
Source pointer
Stores the source address
Setting DMAi destination pointer
DMA0 destination pointer [Address 002616 to 002416] DAR0
DMA1 destination pointer [Address 003616 to 003416] DAR1
DMA2 destination pointer [Address 018616 to 018416] DAR2
DMA3 destination pointer [Address 019616 to 019416] DAR3
(b23) (b19) (b16)(b15) (b8)
b7 b3 b0 b7 b0 b7 b0
Destination pointer
Stores the destination address
Setting DMAi transfer counter
DMA0 transfer counter [Address 002916, 002816] TCR0
DMA1 transfer counter [Address 003916, 003816] TCR1
DMA2 transfer counter [Address 018916, 018816] TCR2
DMA3 transfer counter [Address 019916, 019816] TCR3
(b15) (b8)
b0 b0 b7 b0
Transfer counter
Set a value one less than the transfer count
Setting DMAi control register
b7 b0
1
DMAi control register [Address 002C16, 003C16, 018C16, 019C16]
DMiCON(i = 0 to 3)
DMA enable bit
1 : Enabled
Note: Clear DMA request bit simultaneously again.
When software DMA request bit = "1"
Start DMA transmission

Figure 2.10.8. Set-up procedure of repeated transfer mode

2.11 CRC Calculation Circuit

2.11.1 Overview

Cyclic Redundancy Check (CRC) is a method that compares CRC code formed from transmission data by use of a polynomial generation with CRC check data so as to detect errors in transmission data. Using the CRC calculation circuit allows generation of CRC code. The microcomputer uses a generator polynomial of CRC_CCITT ($X^{16} + X^{12} + X^5 + 1$) or CRC-16 ($X^{16} + X^{15} + X^2 + 1$) to generte CRC code.

And, the CRC circuit includes the ability to snoop reads and writes to SFR addresses. This can be used to accumulate the CRC value on a stream of data without using extra bandwidth to explicitly write data into the CRCIN register.

(1) Registers related to CRC calculation circuit

Figure 2.11.1 shows the memory map of CRC-related registers, and Figure 2.11.2 shows CRC- related registers.

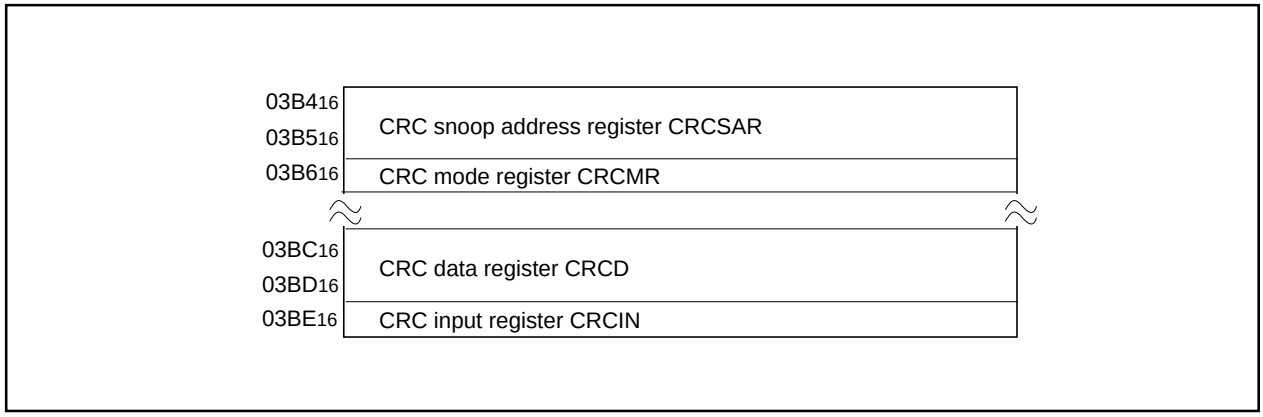
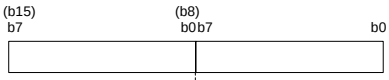


Figure 2.11.1. Memory map of CRC-related registers

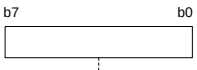
CRC data register



Symbol	Address	When reset
CRCD	03BC16 to 03BD16	Indeterminate

Function	Values that can be set	R	W
CRC calculation result output	000016 to FFFF16	O	O

CRC input register



Symbol	Address	When reset
CRCIN	03BE16	Indeterminate

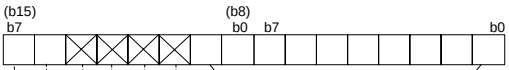
Function	Values that can be set	R	W
Data input	0016 to FF16	O	O

CRC mode register

Symbol	Address	When reset
CRCMR	03B616	0016

Bit symbol	Bit name	Function	R	W
CRCPS	CRC mode polynomial selection bit	0 : $X^{16}+X^{12}+X^5+1$ (CRC-CCITT) 1 : $X^{16}+X^{15}+X^2+1$ (CRC-16)	O	O
Nothing is assigned. Write “0” when writing to this bit. The value is “0” if read.			–	–
CRCMS	CRC mode mode selection bit	0 : LSB first mode 1 : MSB first mode	O	O

CRC snoop address register



Symbol	Address	When reset
CRCSAR	03B516, 03B416	00XXXX?? ????????2

	Bit name	Function	R	W
CRCSAR9-0	CRC Snoop address bits	SFR address to snoop (Note)	O	O
Nothing is assigned. Write “0” when writing to this bit. The value is indeterminate if read.			–	–
CRCSR	CRC Snoop on read enable bit	0 : Disabled 1 : Enabled	O	O
CRCSW	CRC Snoop on write enable bit	0 : Disabled 1 : Enabled	O	O

Note: Only USB, UART and SSI related registers can be snooped.

Figure 2.11.2. CRC-related registers

2.11.2 Operation of CRC Calculation Circuit

The following describes the operation of the CRC calculation. Figure 2.11.3 shows an example of calculation using the CRC-CCITT.

Operation (1) Select CRC-CCITT or CRC-16, and LSB first or MSB first by bit 0 and bit 5 of CRC mode register.
(2) The CRC calculation circuit sets an initial value in the CRC data register.
(3) Writing 1 byte data to the CRC input register generates CRC code based on the data register. CRC code generation for 1 byte data finishes in two machine cycles.
(4) When several bytes of CRC calculation is performed in succession, write the following data in the CRC input register continuously.
(5) The content of CRC data register after all data is written becomes CRC code.

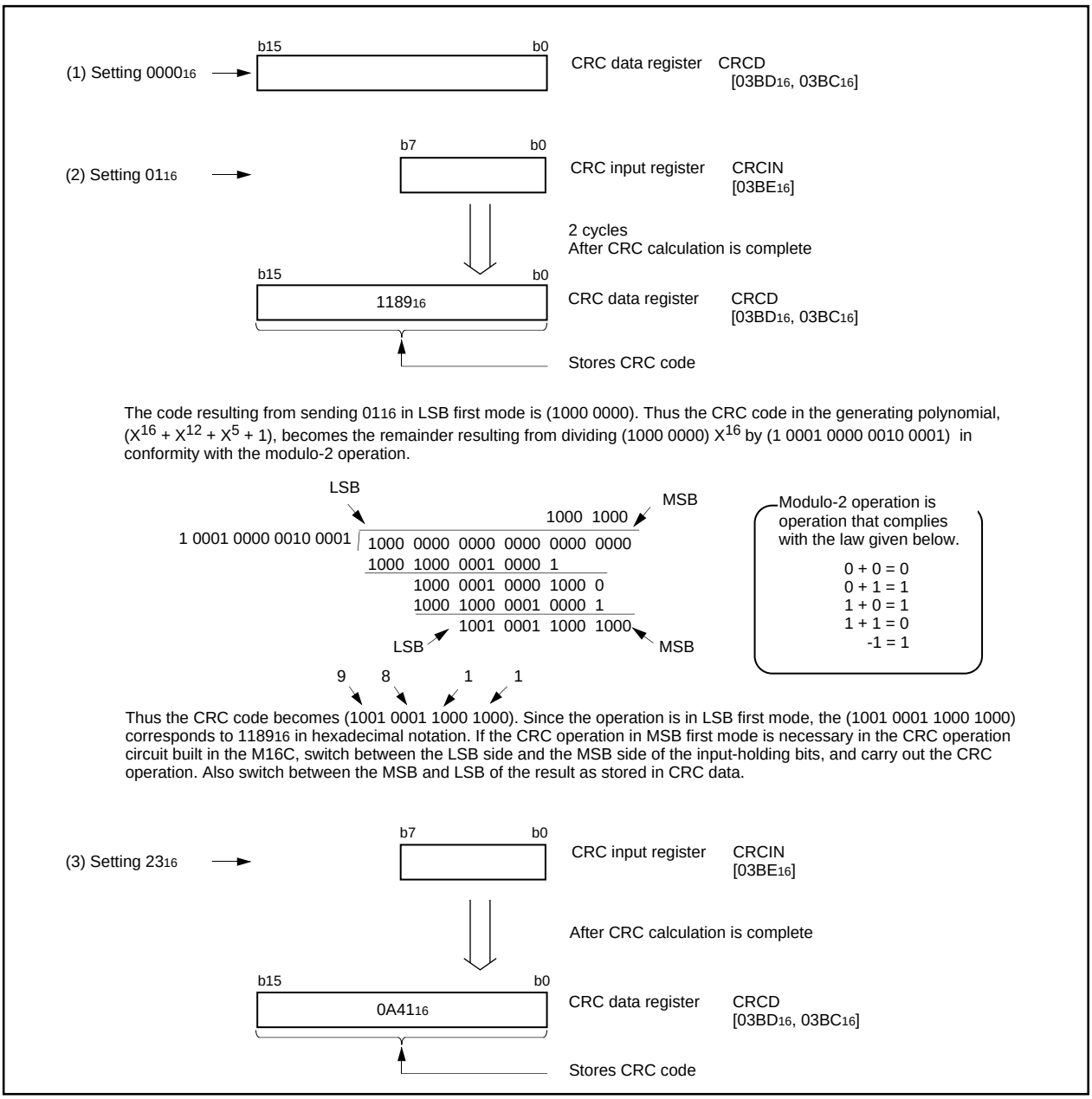

(1) Setting 0000₁₆
b15
b0
CRC data register CRCD [03BD₁₆, 03BC₁₆]
(2) Setting 01₁₆
b7
b0
CRC input register CRCIN [03BE₁₆]
2 cycles
After CRC calculation is complete
b15
b0
1189₁₆
CRC data register CRCD [03BD₁₆, 03BC₁₆]
Stores CRC code
The code resulting from sending 01₁₆ in LSB first mode is (1000 0000). Thus the CRC code in the generating polynomial, (X16 + X12 + X5 + 1), becomes the remainder resulting from dividing (1000 0000) X16 by (1 0001 0000 0010 0001) in conformity with the modulo-2 operation.
LSB
MSB
1000 1000
1 0001 0000 0010 0001
1000 0000 0000 0000 0000 0000
1000 1000 0001 0000 1
1000 0001 0000 1000 0
1000 1000 0001 0000 1
1001 0001 1000 1000
LSB
MSB
9 8 1 1
Modulo-2 operation is operation that complies with the law given below.
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0
-1 = 1
Thus the CRC code becomes (1001 0001 1000 1000). Since the operation is in LSB first mode, the (1001 0001 1000 1000) corresponds to 1189₁₆ in hexadecimal notation. If the CRC operation in MSB first mode is necessary in the CRC operation circuit built in the M16C, switch between the LSB side and the MSB side of the input-holding bits, and carry out the CRC operation. Also switch between the MSB and LSB of the result as stored in CRC data.
(3) Setting 23₁₆
b7
b0
CRC input register CRCIN [03BE₁₆]
After CRC calculation is complete
b15
b0
0A41₁₆
CRC data register CRCD [03BD₁₆, 03BC₁₆]
Stores CRC code

Figure 2.11.3. Calculation example using the CRC calculation circuit (when using CRC-CCITT)

2.11.3 SFR Access Snoop Function

The CRC calculation circuit includes the ability to snoop write/read to/from the SFR addresses and to execute CRC automatic calculation (SFR access snoop function). In order to execute CRC calculation for data which have been written/read to/from the SFR, setting data to CRC input register again is not required. The target SFRs include USB-related registers, UART-related registers, and Serial Sound Interface-related registers.

Operation (1) The bit 1 of CRC mode register selects either CRC-CCITT or CRC-16, and the bit 7 selects either LSB first or MSB first.

(2) The target SFR addresses are set to CRC snoop address register (bit 0 to 9 of CRCSAR). Snooping of writing to the target SFR is enabled with CRC snoop on write enable bit (bit 15 of CRCSAR), and snooping of reading from the target SFR is enabled with CRC snoop on read enable bit (bit 14 of CRCSAR).

(3) The initial value 0000_{16} is set to CRC data register.

(4) If writing into the target SFR is executed by either the CPU or DMA while “1” is set to CRC snoop on write enable bit, the CRC calculation circuit will store the data written to the target SFR in CRC input register, and executes CRC calculation. Similarly, if reading from the target SFR by the CPU or DMA while “1” is set to CRC snoop on read enable bit, calculation circuit will store the data read from the target SFR in CRC input register, and executes CRC calculation. The CRC calculation circuit can only calculate CRC codes on data 1-byte at a time. Therefore, if a target SFR is accessed in a word (16-bit) bus cycle, only 1-byte data are stored into CRC input register.

(5) When 1-byte data is stored in CRC input register, CRC codes are generated in CRC data register based on the stored data and the content of CRC data register. Generation of CRC codes for 1-byte of data is completed in two machine cycles.

2.12 Watchdog Timer

2.12.1 Overview

The watchdog timer can detect a runaway program using its 15-bit timer prescaler. The following is an overview of the watchdog timer.

(1) Watchdog timer start procedure

When reset, the watchdog timer is in stopped state. Writing to the watchdog timer start register initializes the watchdog timer to $7FFF_{16}$ and causes it to start performing a down count. The watchdog timer, once started operating, cannot be stopped by any means other than stopping conditions.

(2) Watchdog timer stop conditions

The watchdog timer stops in any one of the following states:

(a) Period in which the CPU is in stopped state

(b) Period in which the CPU is in waiting state

(c) Period in which the microcomputer is in hold state

(3) Watchdog timer initialization

The watchdog timer is initialized to $7FFF_{16}$ in the cases given below, and begins a down count.

(a) When the watchdog timer writes to the watchdog timer start register while a count is in progress

(b) When the watchdog timer underflows

(4) Runaway detection

When the watchdog timer underflows, either a watchdog timer interrupt occurs or reset is selected depending on the setting of the watchdog timer function select bit. In writing a program, write to the watchdog timer start register before the watchdog timer underflows.

The watchdog timer interrupt occurs regardless of the status of the interrupt enable flag (I flag). In processing a watchdog timer interrupt, set the software reset bit to “1” to reset software.

(5) Watchdog timer cycle

The watchdog timer cycle varies depending on the BCLK and the frequency division ratio of the prescaler selected.

Table 2.12.1 shows the watchdog timer cycle.

Table 2.12.1. The watchdog timer cycle (f(XIN) = 16MHz)

CM07	CM06	CM17	CM16	BCLK	WDC7	Period
0	0	0	0	16MHz	0	Approx. 32.8ms (Note)
					1	Approx. 262.1ms (Note)
0	0	0	1	8MHz	0	Approx. 65.5ms (Note)
					1	Approx. 524.3ms (Note)
0	0	1	0	4MHz	0	Approx. 131.1ms (Note)
					1	Approx. 1.049s (Note)
0	0	1	1	1MHz	0	Approx. 524.3ms (Note)
					1	Approx. 4.194s (Note)
0	1	Invalid	Invalid	2MHz	0	Approx. 262.1ms (Note)
					1	Approx. 2.097s (Note)
1	Invalid	Invalid	Invalid	32kHz	Invalid	Approx. 2s (Note)

Note: An error due to the prescaler occurs.

(6) Registers related to the watchdog timer

Figure 2.12.1 shows the memory map of watchdog timer-related registers, and Figure 2.12.2 shows watchdog timer-related registers.

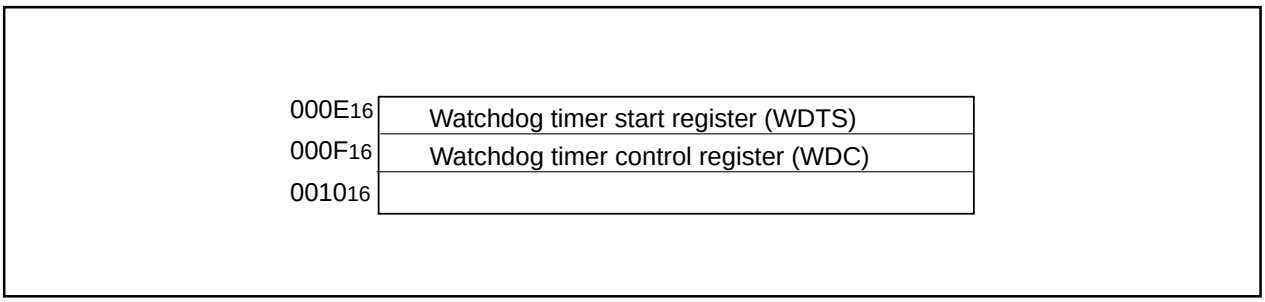


Figure 2.12.1. Memory map of watchdog timer-related registers

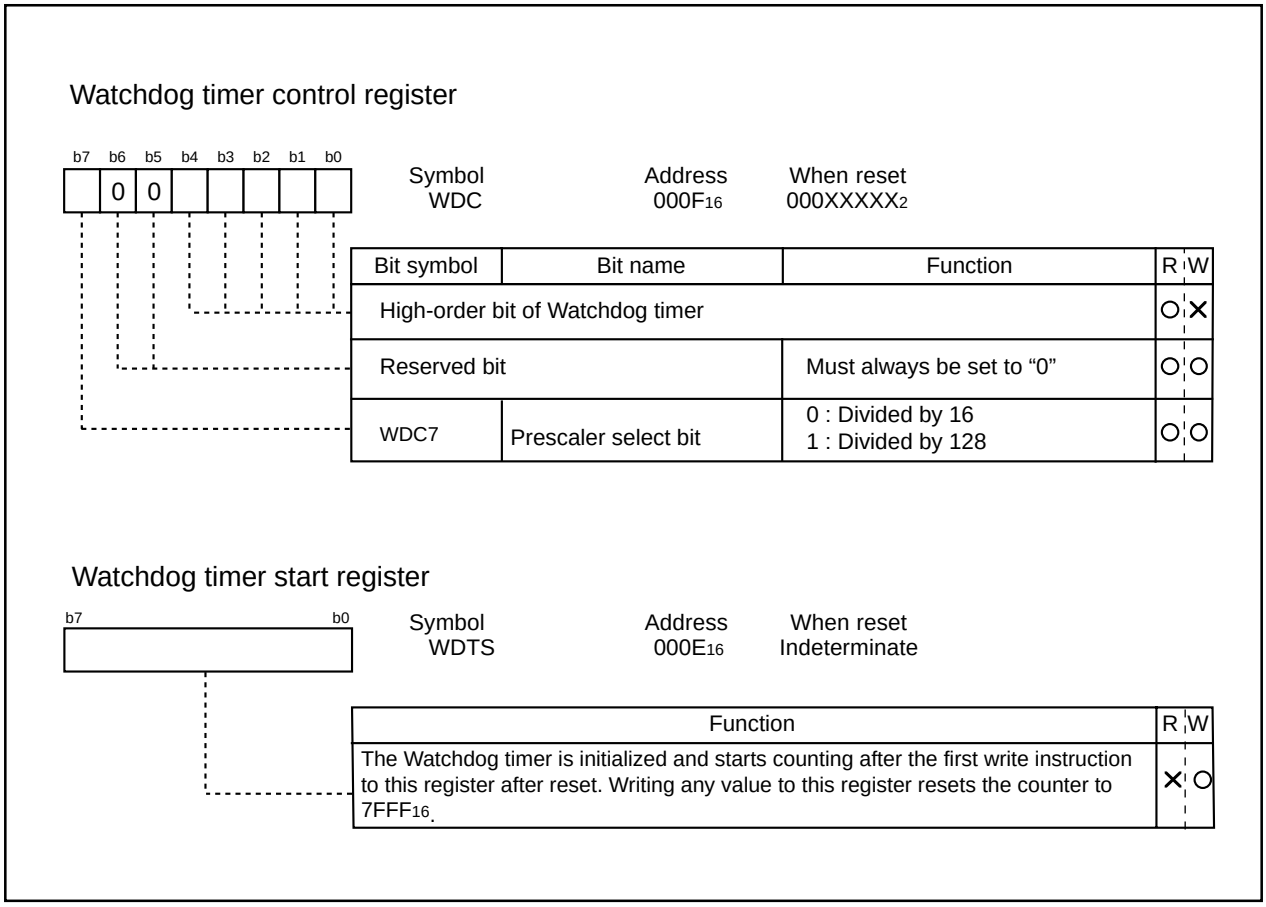

Watchdog timer control register
b7 b6 b5 b4 b3 b2 b1 b0
0 0
Symbol WDC
Address 000F16
When reset 000XXXXX2
Bit symbol | Bit name | Function | R | W
High-order bit of Watchdog timer | O | X
Reserved bit | Must always be set to "0" | O | O
WDC7 | Prescaler select bit | 0 : Divided by 16 1 : Divided by 128 | O | O
Watchdog timer start register
b7 b0
Symbol WDTS
Address 000E16
When reset Indeterminate
Function | R | W
The Watchdog timer is initialized and starts counting after the first write instruction to this register after reset. Writing any value to this register resets the counter to 7FFF16. | X | O

Figure 2.12.2. Watchdog timer-related registers

2.12.2 Operation of Watchdog Timer (Watchdog timer interrupt)

The following is an operation of the watchdog timer using watchdog timer interrupt. Figure 2.12.3 shows the operation timing, and Figure 2.12.4 shows the set-up procedure.

Operation (1) Writing to the watchdog timer start register initializes the watchdog timer to 7FFF16 and causes it to start a down count.

(2) With a count in progress, writing to the watchdog timer start register again initializes the watchdog timer to 7FFF16 and causes it to resume counting.

(3) Either executing the WAIT instruction or going to the stopped state causes the watchdog timer to hold the count in progress and to stop counting. The watchdog timer resumes counting after returning from the execution of the WAIT instruction or from the stopped state.

(4) If the watchdog timer underflows, it is initialized to 7FFF16 and continues counting. At this time, a watchdog timer interrupt occurs.

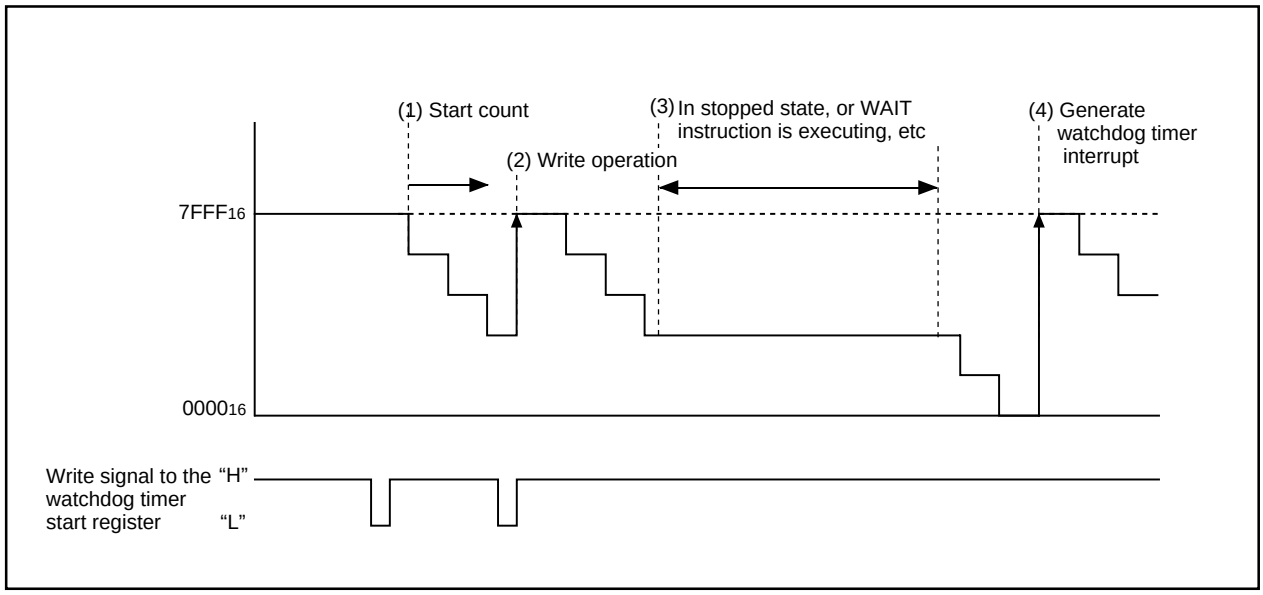


Figure 2.12.3. Operation timing of watchdog timer (watchdog timer interrupt)

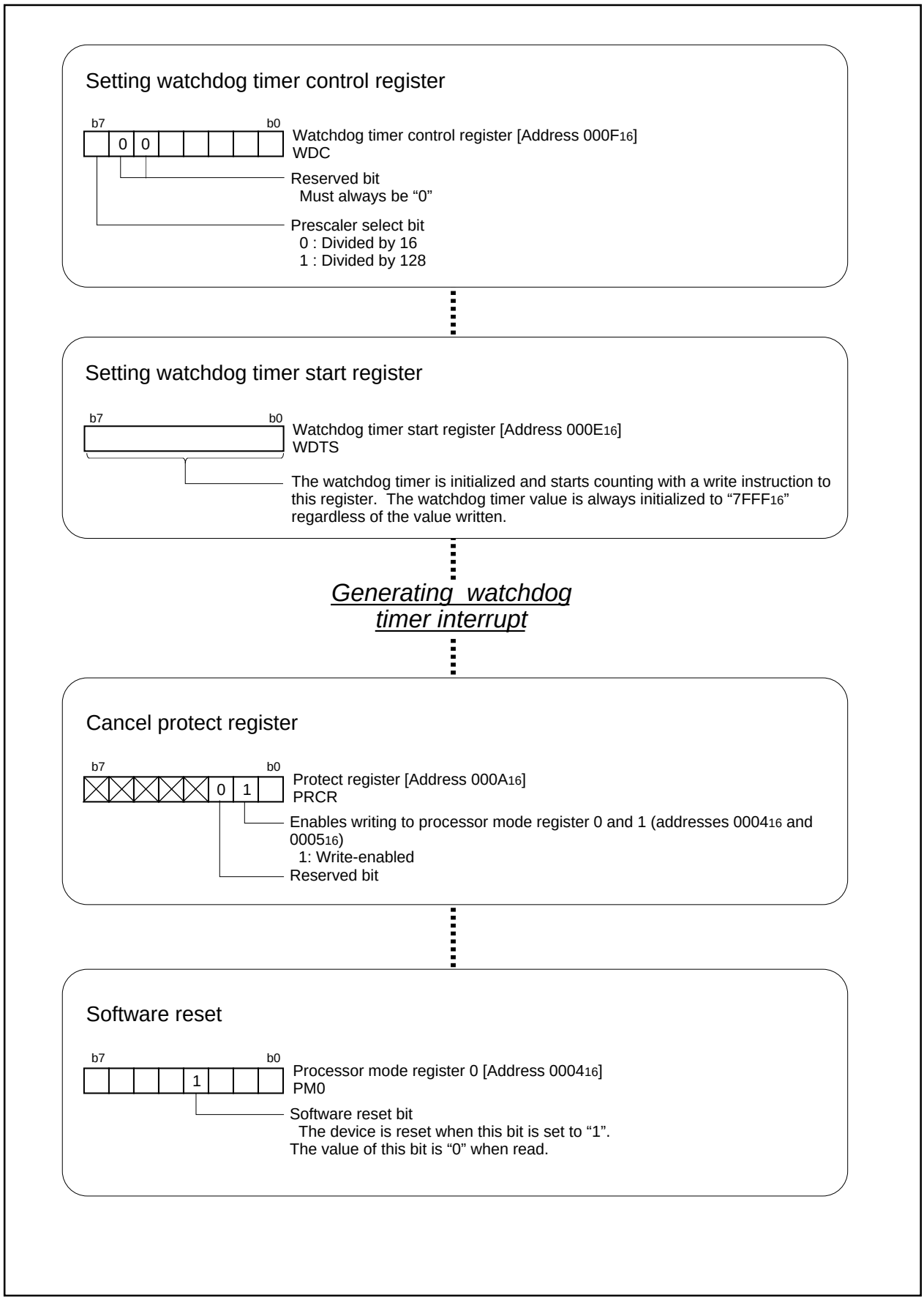


Figure 2.12.4. Set-up procedure of watchdog timer interrupt (watchdog timer interrupt)

2.13 Address Match Interrupt Usage

2.13.1 Overview of the address match interrupt usage

The address match interrupt is used for correcting a ROM or for a simplified debugging-purpose monitor. When the external bus is used for 8 bits, the address match interrupt is not used to the external areas. The following is an overview of the address match interrupt usage.

(1) Enabling/disabling the address match interrupt

The address match interrupt enable bit can be used to enable and disable an address match interrupt. It is affected neither by the processor interrupt priority level (IPL) nor the interrupt enable flag (I flag).

(2) Timing of the address match interrupt

An interrupt occurs immediately before executing the instruction in the address indicated by the address match interrupt register. Set the first address of the instruction in the address match interrupt register. Setting a half address of an instruction or an address of tabulated data does not generate an address match interrupt.

The first instruction of an interrupt routine does not generate an address match interrupt either.

(3) Returning from an address match interrupt

The address put in the stack when an address match interrupt occurs depends on the instruction not yet executed (the instruction the address match interrupt register indicates). The return address is not put in the stack. For this reason, to return from an address match interrupt, either rewrite the content of the stack and use the REIT instruction or use the POP instruction to restore the stack to the state as it was before the interrupt occurred and return by use of a jump instruction.

<Instructions whose address is added to by 2 when an address match interrupt occurs>

- 16-bit operation code instructions
- 8-bit operation code instructions given below

ADD.B:S	#IMM8,dest	SUB.B:S	#IMM8,dest	AND.B:S	#IMM8,dest
OR.B:S	#IMM8,dest	MOV.B:S	#IMM8,dest	STZ.B:S	#IMM8,dest
STNZ.B:S	#IMM8,dest	STZX.B:S	#IMM81,#IMM82,dest		
CMP.B:S	#IMM8,dest	PUSHM	src	POPM	dest
JMPS	#IMM8	JSRS	#IMM8		
MOV.B:S	#IMM,dest (However, dest = A0/A1)				

<Instructions whose address is added to by 1 when an address match interrupt occurs>

- Instructions other than those listed above

Figure 2.13.1. Unexecuted instructions and corresponding stacked addresses

Figure 2.13.1 shows unexecuted instructions and corresponding the stacked addresses.

(4) How to determine an address match interrupt

Address match interrupts can be set at two different locations. However, both location will have the same vector address. Therefore, it is necessary to determine which interrupt has occurred; address match interrupt 0 or address match interrupt 1. Using the content of the stack, etc., determine which interrupt has occurred according to the first part of the address match interrupt routine.

(5) Registers related to the address match interrupt

Figure 2.13.2 shows the memory map of address match interrupt-related registers, and Figure 2.13.3 shows address match interrupt-related registers.

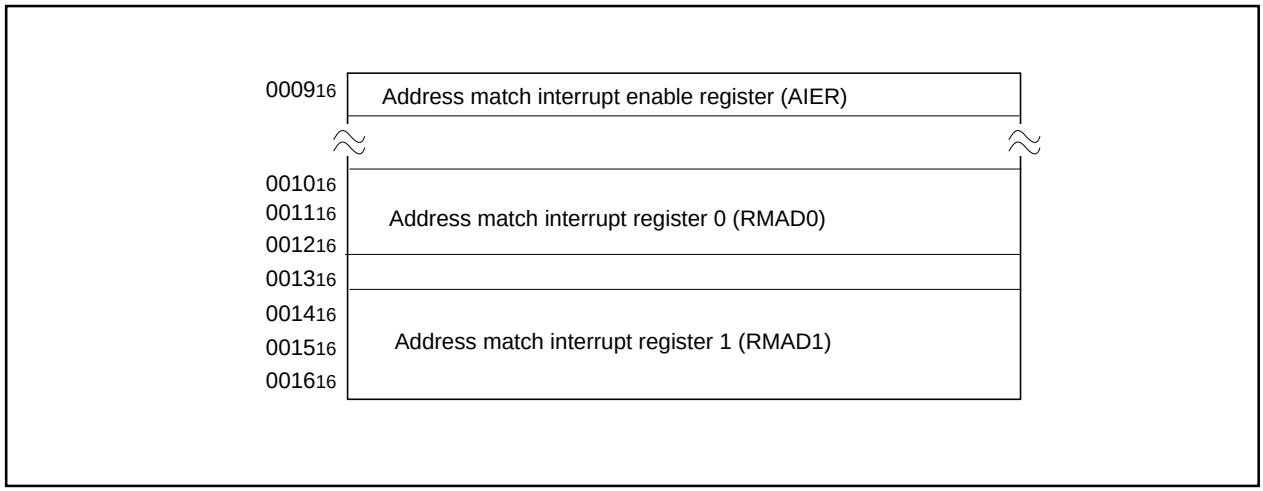


Figure 2.13.2. Memory map of address match interrupt-related registers

Address match interrupt enable register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
AIER	0009₁₆	XXXXXX00₂

Bit symbol	Bit name	Function	R	W
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disaled 1 : enabled	○	○
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disaled 1 : enabled	○	○
Nothing is assigned. Write “0” when writing to these bits. If read, the value is indeterminate.			—	—

Address match interrupt register i (i = 0, 1)

(b23) b7	(b19) b3	(b16) b0	(b15) b7	(b8) b0	b7	b0

Symbol	Address	When reset
RMAD0	0012₁₆ to 0010₁₆	X00000₁₆
RMAD1	0016₁₆ to 0014₁₆	X00000₁₆

Function	Values that can be set	R	W
Address setting register for address match interrupt	00000₁₆ to FFFFF₁₆	○	○
Nothing is assigned. Write “0” when writing to these bits. If read, the value is indeterminate.		—	—

Figure 2.13.3. Address match interrupt-related registers

2.13.2 Operation of Address Match Interrupt

The following is an operation of address match interrupt. Figure 2.13.4 shows the set-up procedure of address match interrupt, and Figure 2.13.5 shows the overview of the address match interrupt handling routine.

Operation (1) The address match interrupt handling routine sets an address to be used to cause the address match interrupt register to generate an interrupt.
(2) Setting the address match enable flag to “1” enables an interrupt to occur.
(3) An address match interrupt occurs immediately before the instruction in the address indicated by the address match interrupt register as a program is executed.

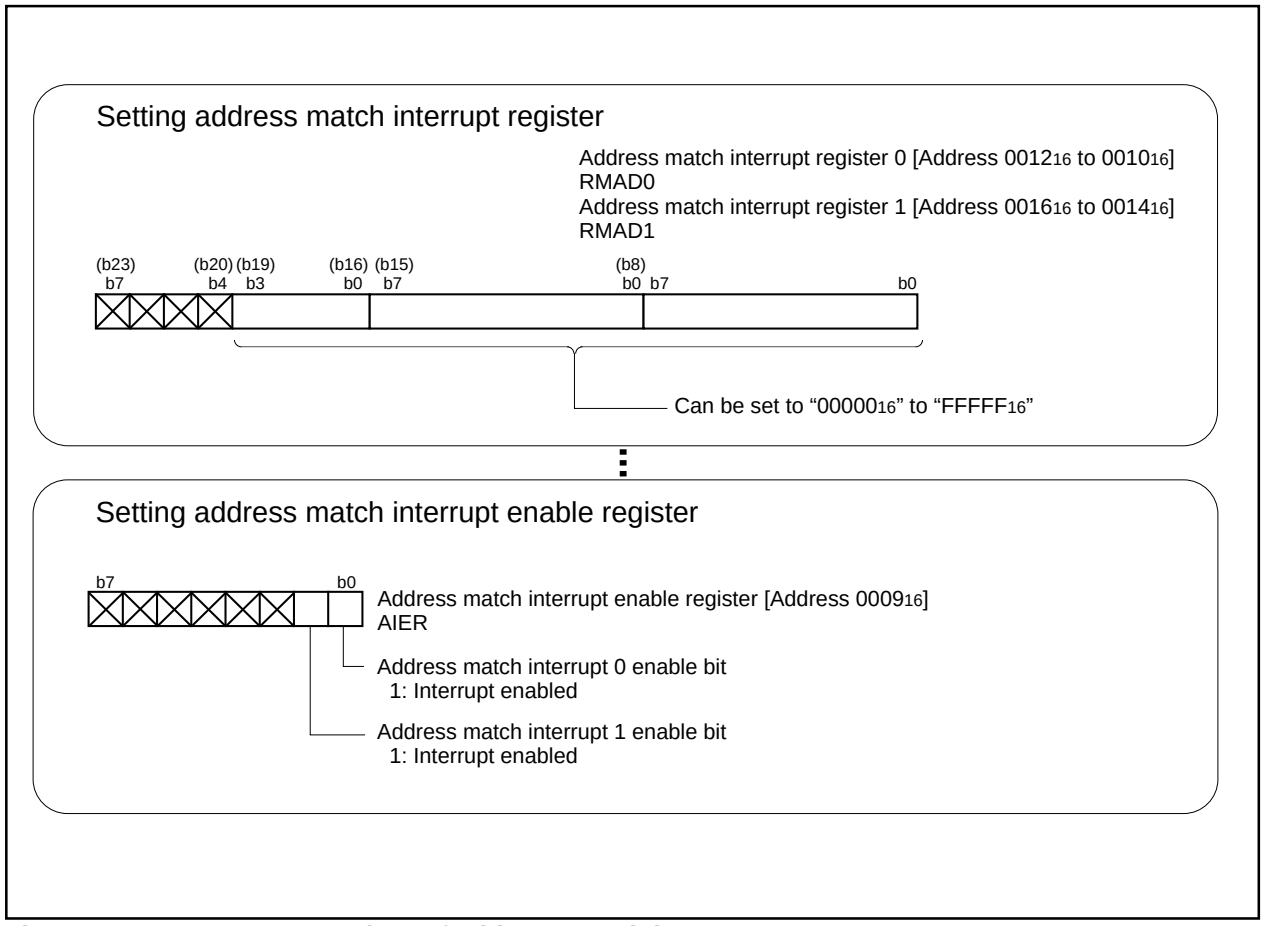
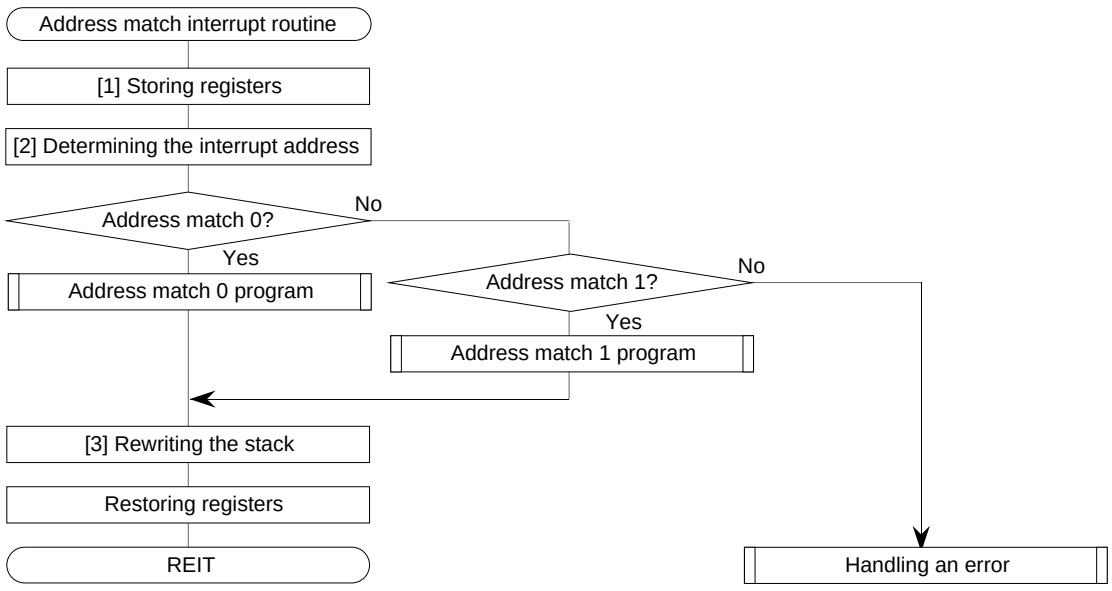

Setting address match interrupt register
Address match interrupt register 0 [Address 0012₁₆ to 0010₁₆]
RMAD0
Address match interrupt register 1 [Address 0016₁₆ to 0014₁₆]
RMAD1
(b23) (b20) (b19) (b16) (b15) (b8)
b7 b4 b3 b0 b7 b0 b7 b0
Can be set to “00000₁₆” to “FFFFF₁₆”
Setting address match interrupt enable register
b7 b0
Address match interrupt enable register [Address 0009₁₆]
AIER
Address match interrupt 0 enable bit
1: Interrupt enabled
Address match interrupt 1 enable bit
1: Interrupt enabled

Figure 2.13.4. Set-up procedure of address match interrupt



Explanation:

[1] Storing the contents of the registers holding the main program status to be kept.

[2] Determining the interrupt address
Determining which factor generated the interrupt.

[3] Rewriting the stack
Rewriting the return address.

Figure 2.13.5. Overview of the address match interrupt handling routine

2.14 Key-Input Interrupt Usage

2.14.1 Overview of the key-input interrupt usage

Key-input interrupt can be generated by a falling edge, rising edge or both edges input to any Port 10 pin. It can also be used as a Key-on wake up function for canceling the wait mode or stop mode. It is possible to select the edge of the Key input interrupt for P10 with bits 0 and 1 of key input mode register. This register is also used to enable or disable Port 10 pins that are to be used for Key-input interrupts. Port 10 can be configured with pull-up resistors using the pull-up control resistor.
The following is an overview of the key-input interrupt usage:

(1) Enabling/disabling the key-input interrupt

The key-input interrupt can be enabled and disabled using the key-input mode register ($03F9_{16}$) and the key-input interrupt register (0041_{16}). The key-input interrupt is affected by the interrupt priority level (IPL) and the interrupt enable flag (I flag). A falling edge, rising edge or both edges input to any Port 10 pin can be selected by P10 Key-input edge select bits (bit0 and bit1 of $03F9_{16}$).

(2) Occurrence timing of the key-input interrupt

With key-input interrupt acceptance enabled, pins $P10_0$ through $P10_7$, which are set to input, become key-input interrupt pins ($\overline{KI_0}$ through $\overline{KI_7}$). A Key-input interrupt occurs when the selected edge is input to a Key-input interrupt pin. At this moment, the level of other key-input interrupt pins must be “H”. No interrupt occurs when the level of other key-input interrupt pins is “L”.

(3) How to determine a key-input interrupt

A key-input interrupt occurs when the selected edge is input to one of eight pins, but each pin has the same vector address. Therefore, read the input level of Port P10 in the key-input interrupt routine to determine the interrupted pin.

(4) Registers related to the key-input interrupt

Figure 2.14.1 shows the memory map of key-input interrupt-related registers, and Figure 2.14.2 and 2.14.3 show key-input interrupt-related registers.

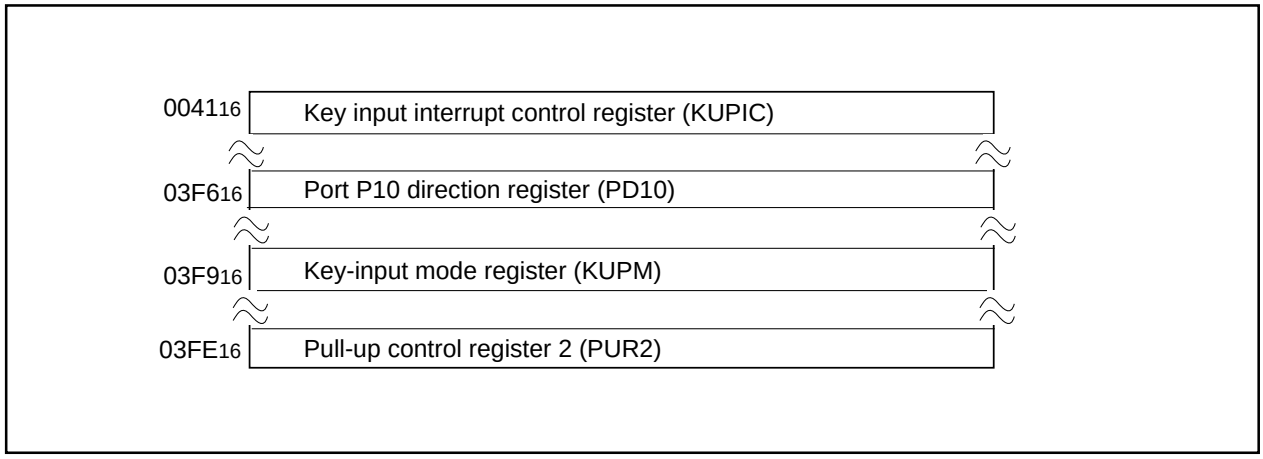
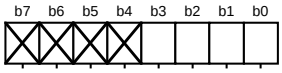


Figure 2.14.1. Memory map of key-input interrupt-related registers

Key input interrupt control register (Note 1)

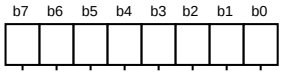


Symbol	Address	When reset
KUPIC	0041₁₆	XXXXX000₂

Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1: Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note 2)
Nothing is assigned. Write “0” when writing to these bits. The contents are indeterminate if read.			–	–

Note 1: To rewrite the interrupt control register, do so at a point that dose not generate the interrupt request for that register. For details, see the precautions for interrupts.
Note 2: This bit can only be reset (=0), but cannot be set (=1).

Port P10 direction register



Symbol	Address	When reset
PD10	03F6₁₆	00₁₆

Bit symbol	Bit symbol	Bit symbol	R	W
PD10_0	Port P10₀ direction register	0 : Imput mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD10_1	Port P10₁ direction register		○	○
PD10_2	Port P10₂ direction register		○	○
PD10_3	Port P10₃ direction register		○	○
PD10_4	Port P10₄ direction register		○	○
PD10_5	Port P10₅ direction register		○	○
PD10_6	Port P10₆ direction register		○	○
PD10_7	Port P10₇ direction register		○	○

Figure 2.14.2. key-input interrupt-related registers (1)

Key-input mode register

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕						

Symbol	Address	When reset
KUPM	03F9$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	R	W
KIS0	P10 Key-input edge select 0	b1 b0 0 0 : Falling edge 0 1 : Rising edges 1 0 : Two edge 1 1 : Reserved	O	O
KIS1	P10 Key-input edge select 1		O	O
KIE0	P10$_0$ and P10$_1$ Key-input enable bit	0 : Disabled 1 : Enabled	O	O
KIE1	P10$_2$ and P10$_3$ Key-input enable bit	0 : Disabled 1 : Enabled	O	O
KIE2	P10$_4$ and P10$_5$ Key-input enable bit	0 : Disabled 1 : Enabled	O	O
KIE3	P10$_6$ and P10$_7$ Key-input enable bit	0 : Disabled 1 : Enabled	O	O
Nothing is assigned. Write “0” when writing to these bits. The value is “0” if read.			–	–

Pull-up control register 2

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕			✕			

Symbol	Address	When reset
PUR2	03FE$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PU20	P8$_0$ to P8$_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	O	O
PU21	P8$_4$ to P8$_7$ pull-up (Except P8$_5$)		O	O
PU22	P9$_0$ to P9$_3$ pull-up (Except P9$_1$)		O	O
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be “0”.			–	–
PU24	P10$_0$ to P10$_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	O	O
PU25	P10$_4$ to P10$_7$ pull-up		O	O
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–

Figure 2.14.3. key-input interrupt-related registers (2)

2.14.2 Operation of Key-Input Interrupt

The following is an operation of key-input interrupt. Figure 2.14.4 shows an example of a circuit that uses the key-input interrupt, Figure 2.14.5 shows an example of operation of key-input interrupt, and Figure 2.14.6 shows the setting procedure of key-input interrupt.

Operation (1) Set the direction register of the ports to be changed to key-input interrupt pins to input, and set the pull-up function.

(2) Setting the key-input interrupt control register and setting the interrupt enable flag makes the interrupt-enabled state ready.

(3) If a falling edge is input to either $\overline{KI_0}$ through KI_7, the key-input interrupt request bit goes to "1".

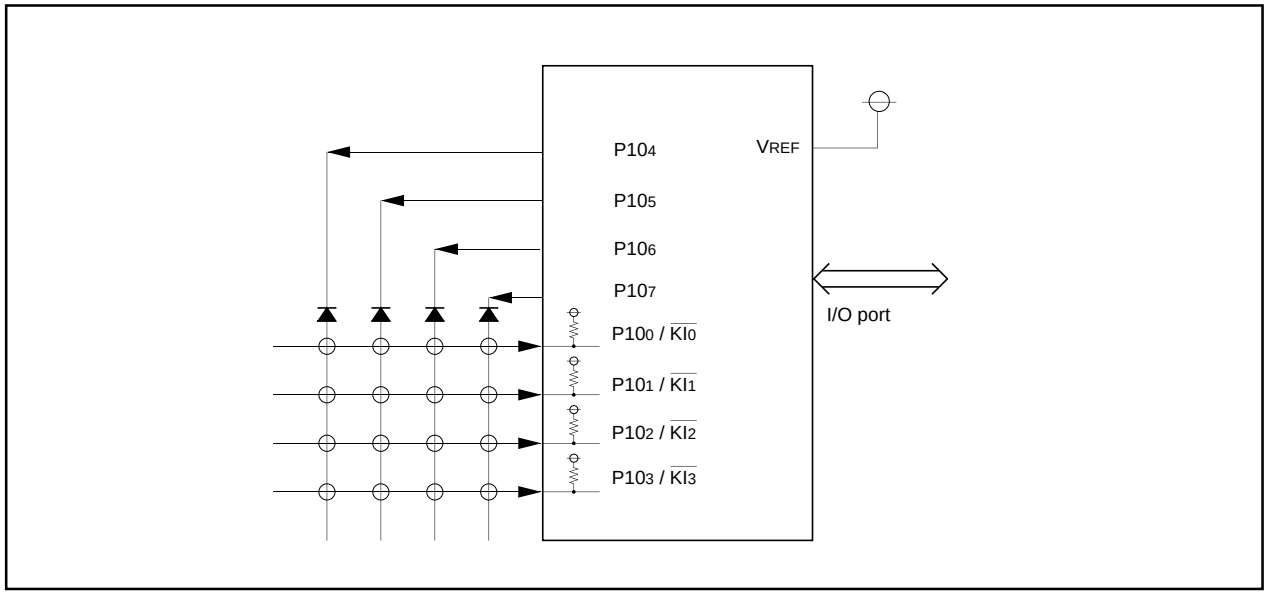


Figure 2.14.4. Example of circuit using the key-input interrupt

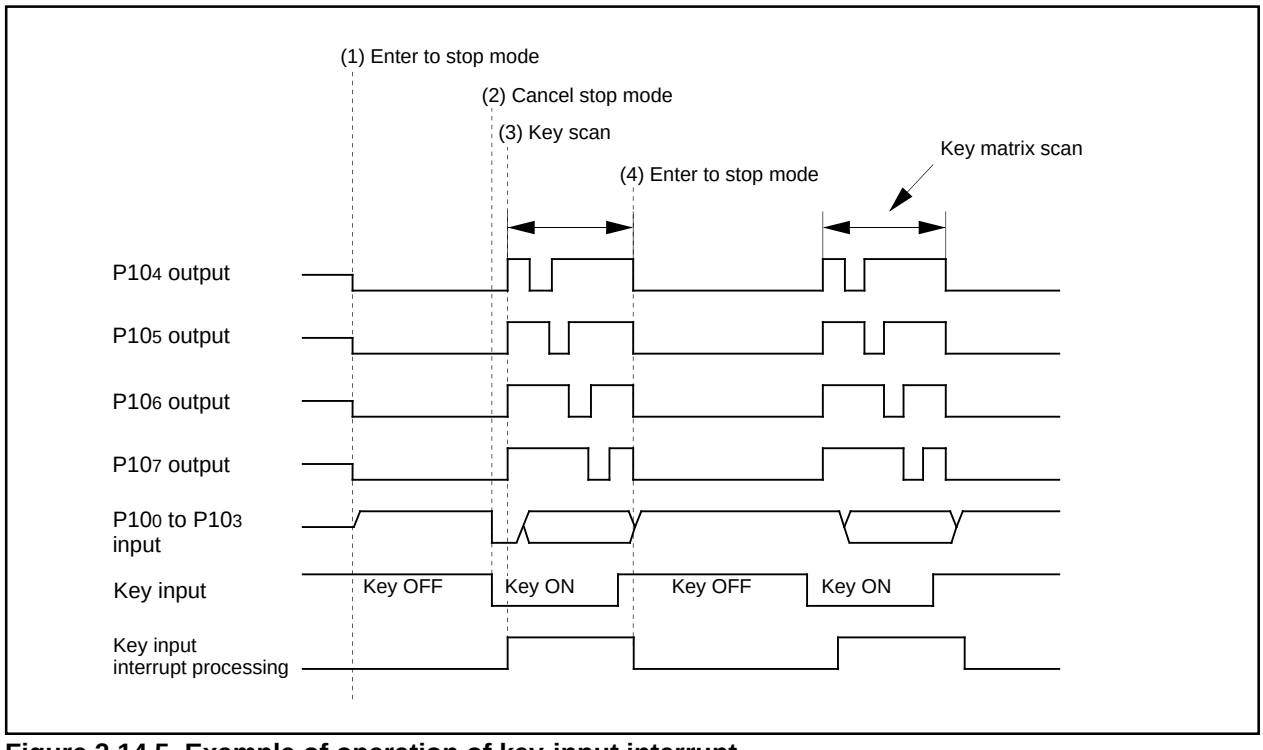


Figure 2.14.5. Example of operation of key-input interrupt

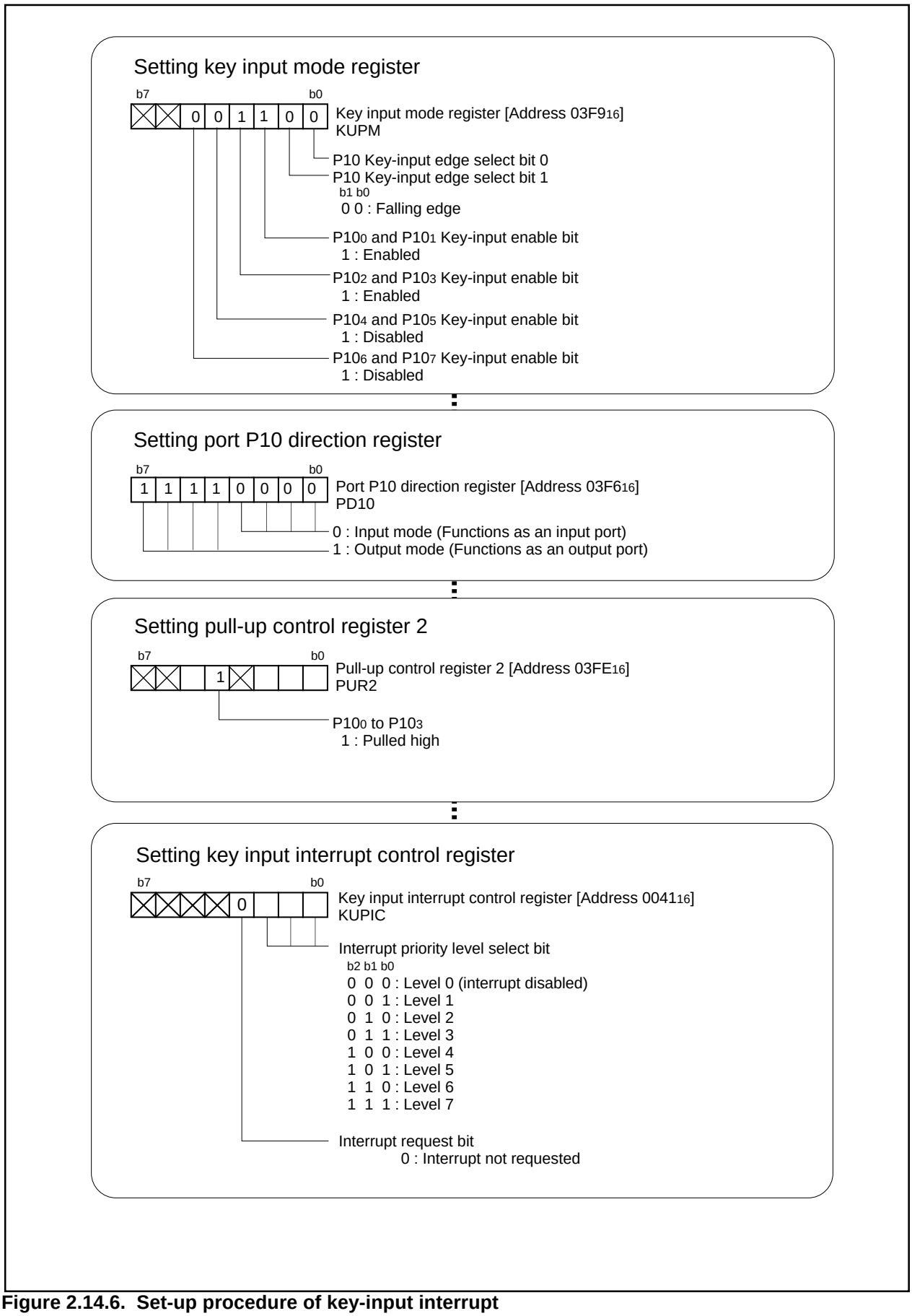

Setting key input mode register
b7 b0
0 0 1 1 0 0
Key input mode register [Address 03F916]
KUPM
P10 Key-input edge select bit 0
P10 Key-input edge select bit 1
b1 b0
0 0 : Falling edge
P100 and P101 Key-input enable bit
1 : Enabled
P102 and P103 Key-input enable bit
1 : Enabled
P104 and P105 Key-input enable bit
1 : Disabled
P106 and P107 Key-input enable bit
1 : Disabled
Setting port P10 direction register
b7 b0
1 1 1 1 0 0 0 0
Port P10 direction register [Address 03F616]
PD10
0 : Input mode (Functions as an input port)
1 : Output mode (Functions as an output port)
Setting pull-up control register 2
b7 b0
1
Pull-up control register 2 [Address 03FE16]
PUR2
P100 to P103
1 : Pulled high
Setting key input interrupt control register
b7 b0
0
Key input interrupt control register [Address 004116]
KUPIC
Interrupt priority level select bit
b2 b1 b0
0 0 0 : Level 0 (interrupt disabled)
0 0 1 : Level 1
0 1 0 : Level 2
0 1 1 : Level 3
1 0 0 : Level 4
1 0 1 : Level 5
1 1 0 : Level 6
1 1 1 : Level 7
Interrupt request bit
0 : Interrupt not requested

Figure 2.14.6. Set-up procedure of key-input interrupt

2.15 Multiple interrupts Usage

2.15.1 Overview of the Multiple interrupts usage

The following is an overview of the multiple interrupts usage.

(1) Interrupt control

Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

Enable or disable a non-maskable interrupt using the interrupt enable flag (I flag), interrupt priority level selection bit, or processor interrupt priority level (IPL). Whether an interrupt request is present or absent is indicated by the interrupt request bit. The interrupt request bit and the interrupt priority level select bit are located in the interrupt control register of each interrupt. Also, the interrupt enable flag (I flag) and the IPL are located in the flag register (FLG).

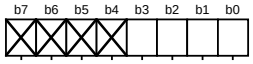
Figure 2.15.1 shows the memory map of the interrupt control registers, and Figure 2.15.2 shows the interrupt control registers.

0041$_{16}$	Key input interrupt control register (KUPIC)
0042$_{16}$	UART2 receive/ACK interrupt control register (S2RIC)
0043$_{16}$	UART1/3 Bus collision interrupt control register (S13BCNIC)
0044$_{16}$	INT1 interrupt control register (INT1IC)
0045$_{16}$	Timer A1 interrupt control register (TA1IC)
0046$_{16}$	USB Endpoint 0 interrupt control register (EP0IC)
0047$_{16}$	Timer A2 interrupt control register (TA2IC)
0048$_{16}$	UART1 receive/ACK/SSI1 interrupt control register (S1RIC)
0049$_{16}$	UART0/2 Bus collision interrupt control register (S02BCNIC)
004A$_{16}$	UART0 receive/ACK/SSI0 interrupt control register (S0RIC)
004B$_{16}$	AD conversion interrupt control register (ADIC)
004C$_{16}$	DMA0 interrupt conrol register (DM0IC)
004D$_{16}$	UART3 transmit/NACK interrupt control register (S3TIC)
004E$_{16}$	DMA1 interrupt control register (DM1IC)
004F$_{16}$	UART2 transmit/NACK interrupt control register (S2TIC)
0050$_{16}$	DMA2 interrupt control register (DM2IC)
0051$_{16}$	UART1 transmit/NACK/SSI1 interrupt control register (S1TIC)
0052$_{16}$	DMA3 interrupt control register (DM3IC)
0053$_{16}$	UART0 transmit/NACK/SSI0 interrupt control register (S0TIC)
0054$_{16}$	Timer A0 interrupt control register (TA0IC)
0055$_{16}$	UART3 receive/ACK interrupt control register (S3RIC)
0056$_{16}$	USB suspend interrupt control register (SUSPIC)
0057$_{16}$	Timer A3 interrupt control register (TA3IC)
0058$_{16}$	USB resume interrupt control register (RSMIC)
0059$_{16}$	Timer A4 interrupt control register (TA4IC)
005A$_{16}$	USB reset interrupt control register (RSTIC)
005B$_{16}$	USB SOF interrupt control register (SOFIC)
005C$_{16}$	USB Vbus detect interrupt control register (VBDIC)
005D$_{16}$	USB function interrupt control register (USBFIC)
005E$_{16}$	INT2 interrupt control register (INT2IC)
005F$_{16}$	INT0 interrupt control register (INT0IC)

Figure 2.15.1. Memory map of the interrupt control registers

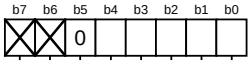
Interrupt control register (Note 1)

Symbol	Address	When reset	Symbol	Address	When reset
KUPIC	0041_{16}	$XXXXX000_2$	S1TIC	0051_{16}	$XXXXX000_2$
S2RIC	0042_{16}	$XXXXX000_2$	DM3IC	0052_{16}	$XXXXX000_2$
S13BCNIC	0043_{16}	$XXXXX000_2$	S0TIC	0053_{16}	$XXXXX000_2$
TA1IC	0045_{16}	$XXXXX000_2$	TA0IC	0054_{16}	$XXXXX000_2$
EP0IC	0046_{16}	$XXXXX000_2$	S3RIC	0055_{16}	$XXXXX000_2$
TA2IC	0047_{16}	$XXXXX000_2$	SUSPIC	0056_{16}	$XXXXX000_2$
S0RIC	$004A_{16}$	$XXXXX000_2$	TA3IC	0057_{16}	$XXXXX000_2$
ADIC	$004B_{16}$	$XXXXX000_2$	RSMIC	0058_{16}	$XXXXX000_2$
DM0IC	$004C_{16}$	$XXXXX000_2$	TA4IC	0059_{16}	$XXXXX000_2$
S3TIC	$004D_{16}$	$XXXXX000_2$	RSTIC	$005A_{16}$	$XXXXX000_2$
DM1IC	$004E_{16}$	$XXXXX000_2$	SOFIC	$005B_{16}$	$XXXXX000_2$
S2TIC	$004F_{16}$	$XXXXX000_2$	VBSIC	$005C_{16}$	$XXXXX000_2$
DM2IC	0050_{16}	$XXXXX000_2$	USBFIC	$005D_{16}$	$XXXXX000_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1: Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note 2)
Nothing is assigned. Write “0” when writing to these bits. The contents are indeterminate if read.			–	–

Symbol	Address	When reset
INT1IC	0044_{16}	$XX00X000_2$
S1RIC	0048_{16}	$XX00X000_2$
S02BCNIC	0049_{16}	$XX00X000_2$
INT2IC	$005E_{16}$	$XX00X000_2$
INT0IC	$005F_{16}$	$XX00X000_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1: Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note 2)
POL	Polarity select bit	0 : Selects falling edge 1 : Selects rising edge	○	○ (Note 3)
Reserved bit		Always set to “0”	○	○
Nothing is assigned. Write “0” when writing to these bits. The contents are indeterminate if read.			–	–

Note 1: To rewrite the interrupt control register, do so at a point that dose not generate the interrupt request for that register. For details, see the precautions for interrupts.
Note 2: This bit can only be reset (=0), but cannot be set (=1).
Note 3: For S1RIC (0048_{16}) and S02BCNIC (0049_{16}), “0” should always be written.

Figure 2.15.2. Interrupt control registers

(2) Interrupt Enable Flag (I flag)

The interrupt enable flag (I flag) controls the enabling and disabling of maskable interrupts. Setting this flag to “1” enables all maskable interrupts; setting it to “0” disables all maskable interrupts. This flag is set to “0” after reset.

The content is changed when the I flag is changed causes the acceptance of the interrupt request in the following timing:

- When changing the I flag using the REIT instruction, the acceptance of the interrupt takes effect as the REIT instruction is executed.
- When changing the I flag using one of the FCLR, FSET, POPC, and LDC instructions, the acceptance of the interrupt is effective as the next instruction is executed.

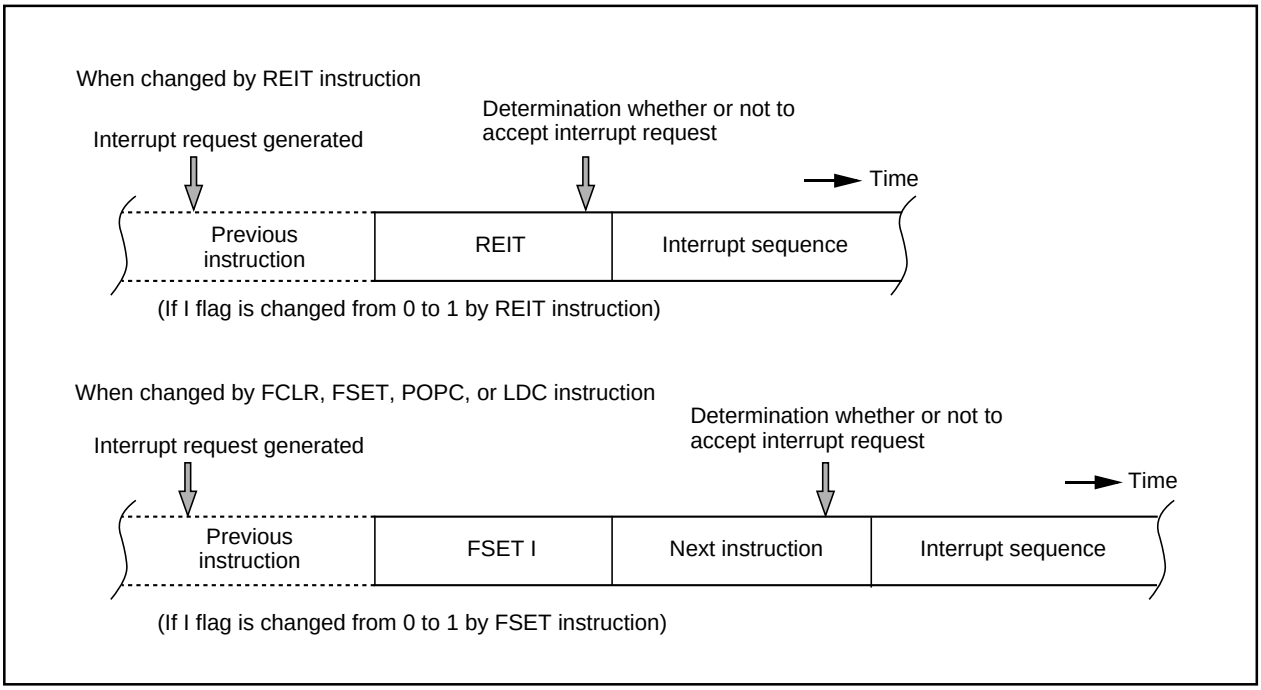

When changed by REIT instruction
Interrupt request generated
Determination whether or not to accept interrupt request
Time
Previous instruction
REIT
Interrupt sequence
(If I flag is changed from 0 to 1 by REIT instruction)
When changed by FCLR, FSET, POPC, or LDC instruction
Interrupt request generated
Determination whether or not to accept interrupt request
Time
Previous instruction
FSET I
Next instruction
Interrupt sequence
(If I flag is changed from 0 to 1 by FSET instruction)

Figure 2.15.3. The timing of reflecting the change in the I flag to the interrupt

(3) Interrupt Request Bit

The interrupt request bit is set to "1" by hardware when an interrupt is requested. After the interrupt is accepted and jumps to the corresponding interrupt vector, the request bit is set to "0" by hardware. The interrupt request bit can also be set to "0" by software. (Do not set this bit to "1").

(4) Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL)

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to “0” disables the interrupt.

Table 2.15.1 shows the settings of interrupt priority levels and Table 2.15.2 shows the interrupt levels enabled, according to the contents of the IPL.

The following are conditions under which an interrupt is accepted:

· interrupt enable flag (I flag) = 1

· interrupt request bit = 1

· interrupt priority level > IPL

The interrupt enable flag (I flag), the interrupt request bit, the interrupt priority select bit, and the IPL are independent, and they are not affected by one another.

Table 2.15.1. Settings of interrupt priority levels

Interrupt priority level select bit	Interrupt priority level	Priority order
b2 b1 b0		
0 0 0	Level 0 (interrupt disabled)	———
0 0 1	Level 1	Low
0 1 0	Level 2	↓
0 1 1	Level 3	↓
1 0 0	Level 4	↓
1 0 1	Level 5	↓
1 1 0	Level 6	↓
1 1 1	Level 7	High

Table 2.15.2. Interrupt levels enabled according to the contents of the IPL

IPL	Enabled interrupt priority levels
IPL_2 IPL_1 IPL_0	
0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

When either the IPL or the interrupt priority level is changed, the new level is reflected to the interrupt in the following timing:

- When changing the IPL using the REIT instruction, the reflection takes effect as of the instruction that is executed in 2 clock cycles after the last clock cycle in volved in the REIT instruction.
- When changing the IPL using either the POPC, LDC or LDIPL instruction, the reflection takes effect as of the instruction that is executed in 3 cycles after the last clock cycle involved in the instruction used.
- When changing the interrupt priority level using the MOV or similar instruction, the reflection takes effect as of the instruction that is executed in 2 clock cycles after the last clock cycle involved in the instruction used.

(5) Interrupt Priority

If there are two or more interrupt requests occurring at a point in time within a single sampling (checking whether interrupt requests are made), the interrupt assigned a higher priority is accepted.

Assign an arbitrary priority to maskable interrupts (peripheral I/O interrupts) using the interrupt priority level select bit. If the same interrupt priority level is assigned, however, the interrupt assigned a higher hardware priority is accepted.

Priorities of the special interrupts, such as Reset (dealt with as an interrupt assigned the highest priority), watchdog timer interrupt, etc. are regulated by hardware.

Figure 2.15.4 shows the priorities of hardware interrupts.

Software interrupts are not affected by the interrupt priority. If an instruction is executed, control branches invariably to the interrupt routine.

Reset > $\overline{\text{NMI}}$ > $\overline{\text{DBC}}$ > Watchdog timer > Peripheral I/O > Single step > Address match

Figure 2.15.4. Hardware interrupts priorities

(6) Interrupt resolution circuit

When two or more interrupts are generated simultaneously, this circuit selects the interrupt with the highest priority level. Figure 2.15.5 shows the circuit that judges the interrupt priority level.

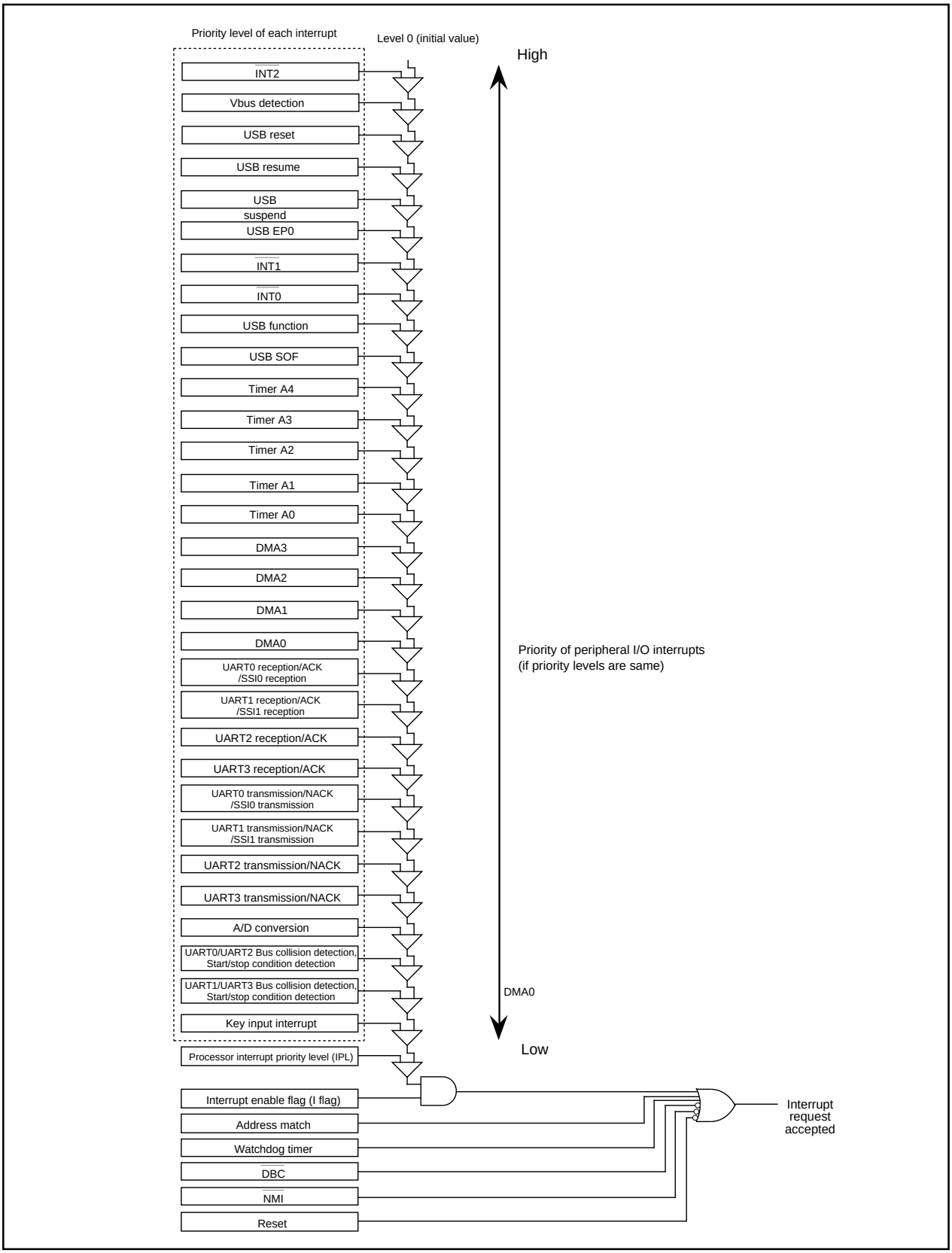


Figure 2.15.5. Interrupts resolution circuit

2.15.2 Multiple Interrupts Operation

The state when control branched to an interrupt routine is described below:

- The interrupt enable flag (I flag) is set to “0” (the interrupt is disabled).
- The interrupt request bit of the accepted interrupt is set to “0”.
- The processor interrupt priority level (IPL) is assigned to the same interrupt priority level as as signed to the accepted interrupt.

Setting the interrupt enable flag (I flag) to “1” within an interrupt routine allows an interrupt request assigned a priority higher than the IPL to be accepted.
An interrupt request that is not accepted because of low priority will be held. If the condition following is met when the REIT instruction returns the IPL and the interrupt priority is determined, then the interrupt request being held is accepted.

Interrupt priority level of the interrupt request being held > Returned the IPL

Figure 2.15.6 shows the example of the multiple interrupts operation.

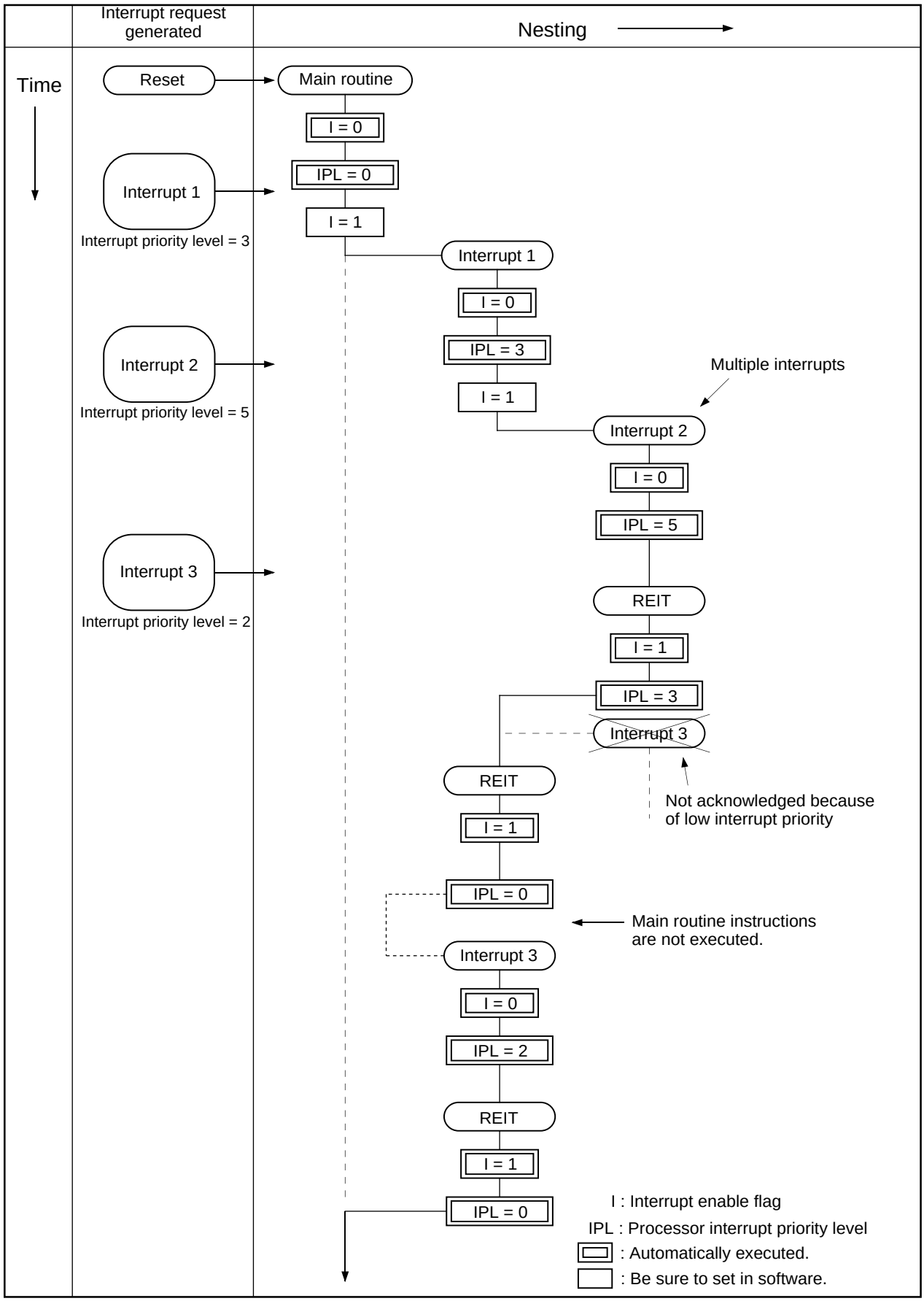


Figure 2.15.6. Example of the multiple interrupts operation

2.16 Power Control Usage

2.16.1 Overview of the power control usage

'Power Control' refers to the reduction of CPU power consumption by stopping the CPU and oscillators, or decreasing the operation clock. The following is a description of the three available power control modes:

(1) Modes

Power control is available in three modes.

(a) Normal operation mode

- **High-speed mode**

  Divide-by-1 frequency of the main clock becomes the BCLK. The CPU operates with the BCLK selected. Each peripheral function operates according to its assigned clock.

- **Medium-speed mode**

  Divide-by-2, divide-by-4, divide-by-8, or divide-by-16 frequency of the main clock becomes the BCLK. The CPU operates according to the BCLK selected. Each peripheral function operates according to its assigned clock.

- **Low-speed mode**

  fc becomes the BCLK. The CPU operates according to the fc clock. The fc clock is supplied by the secondary clock. Each peripheral function operates according to its assigned clock.

- **Low power consumption mode**

  The main clock operating in low-speed mode is stopped. The CPU operates according to the fc clock. The fc clock is supplied by the secondary clock. The only peripheral functions that operate are those with the sub-clock selected as the count source.

(b) Wait mode

The CPU operation is stopped. The oscillators do not stop.

(c) Stop mode

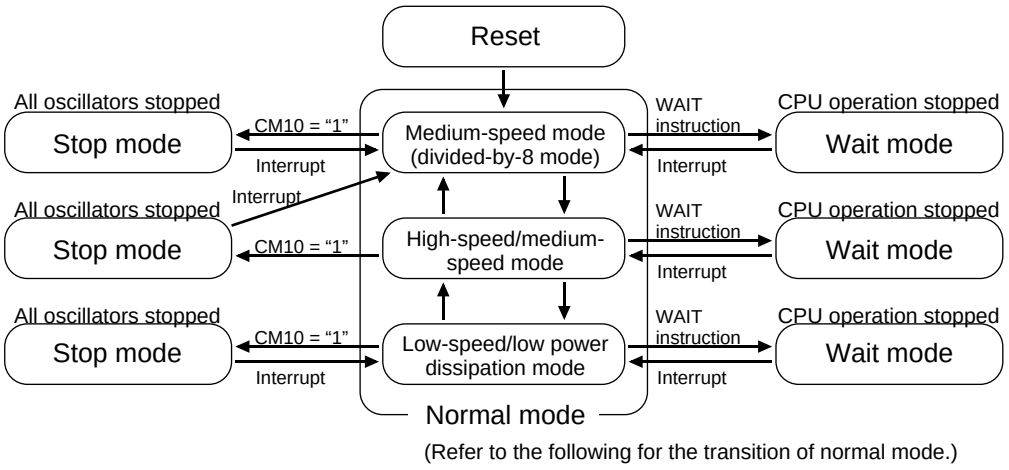
All oscillators stop. The CPU and all built-in peripheral functions stop. This mode, among the three modes listed here, is the most effective in decreasing power consumption.

Figure 2.16.1 is the state transition diagram of the above modes.

(2) Switching the driving capacity of the oscillation circuit

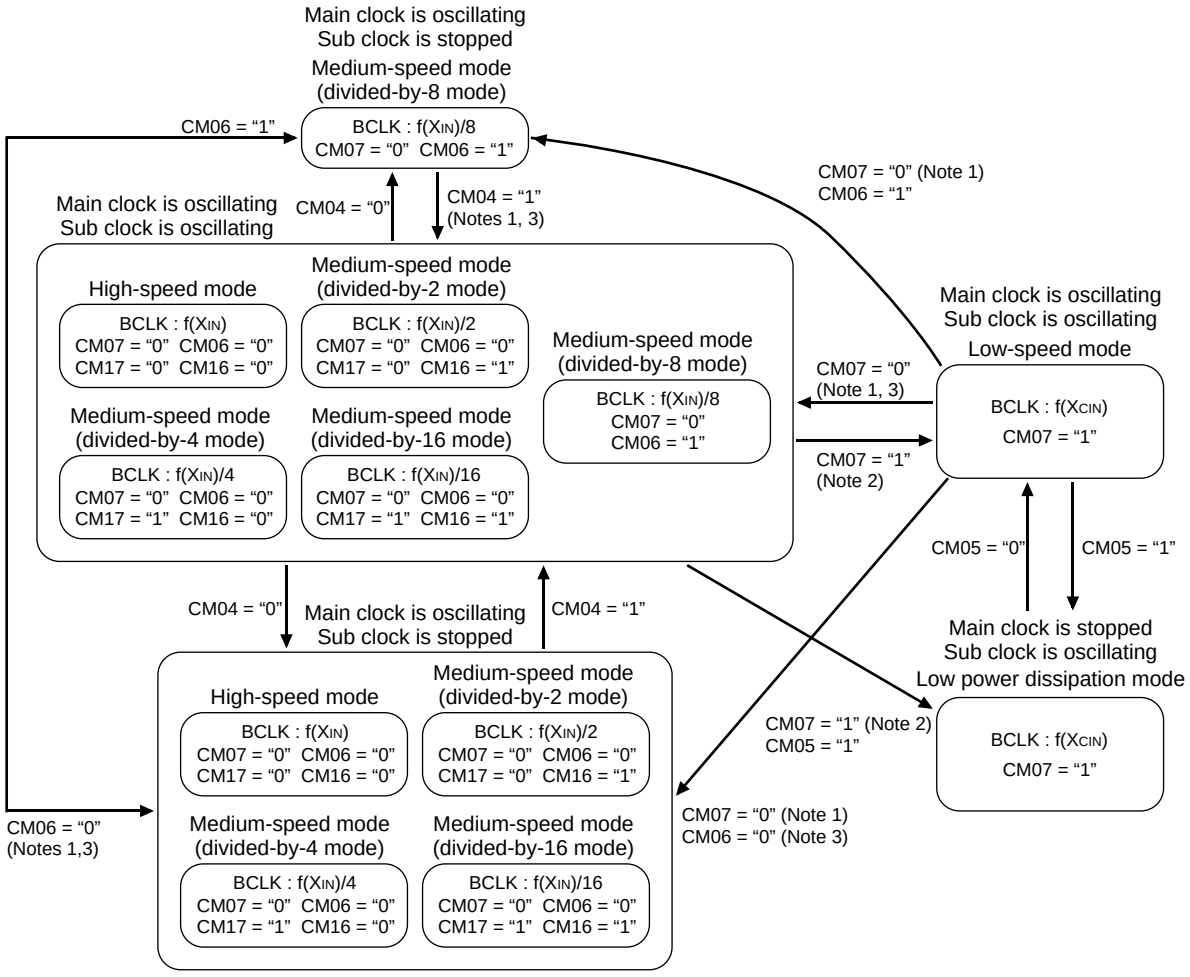
Both the main clock and the secondary clock have the ability to switch the driving capacity.

Transition of stop mode, wait mode



(Refer to the following for the transition of normal mode.)

Transition of normal mode


Main clock is oscillating
Sub clock is stopped
Medium-speed mode (divided-by-8 mode)
BCLK : f(XIN)/8
CM07 = "0" CM06 = "1"
CM06 = "1"
CM07 = "0" (Note 1)
CM06 = "1"
Main clock is oscillating
Sub clock is oscillating
CM04 = "0"
CM04 = "1" (Notes 1, 3)
High-speed mode
BCLK : f(XIN)
CM07 = "0" CM06 = "0"
CM17 = "0" CM16 = "0"
Medium-speed mode (divided-by-2 mode)
BCLK : f(XIN)/2
CM07 = "0" CM06 = "0"
CM17 = "0" CM16 = "1"
Medium-speed mode (divided-by-8 mode)
BCLK : f(XIN)/8
CM07 = "0"
CM06 = "1"
Medium-speed mode (divided-by-4 mode)
BCLK : f(XIN)/4
CM07 = "0" CM06 = "0"
CM17 = "1" CM16 = "0"
Medium-speed mode (divided-by-16 mode)
BCLK : f(XIN)/16
CM07 = "0" CM06 = "0"
CM17 = "1" CM16 = "1"
Main clock is oscillating
Sub clock is oscillating
Low-speed mode
BCLK : f(XCIN)
CM07 = "1"
CM07 = "0" (Note 1, 3)
CM07 = "1" (Note 2)
CM05 = "0"
CM05 = "1"
CM04 = "0"
Main clock is oscillating
Sub clock is stopped
CM04 = "1"
Main clock is stopped
Sub clock is oscillating
Low power dissipation mode
BCLK : f(XCIN)
CM07 = "1"
CM07 = "1" (Note 2)
CM05 = "1"
CM07 = "0" (Note 1)
CM06 = "0" (Note 3)
CM06 = "0"
(Notes 1,3)

Note 1: Switch clock after oscillation of main clock is sufficiently stable.
Note 2: Switch clock after oscillation of sub clock is sufficiently stable.
Note 3: Change CM06 after changing CM17 and CM16.
Note 4: Transit in accordance with arrow.

Figure 2.16.1. State transition diagram of power control mode

(3) Returning from stop mode

The stop mode can be canceled by hardware reset, or by generating an interrupt request. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled and the priority level of the interrupt not to be used for clearing must be set to level 0 before changing to stop mode.

If an interrupt is used to cancel stop mode, that interrupt routine is processed. If only a hardware reset and NMI interrupt are used to cancel stop mode, the priority level of all interrupts must be set to level 0 before changing to stop mode.

When changing from high-speed/medium mode to stop mode and at reset, the main clock division select bit 0 (bit 6 at address 0006_{16}) is set to “1”. When changing from low-speed/low power dissipation mode, the value before stop mode is retained.

(4) Returning form wait mode

The wait mode can be canceled by hardware reset, or by generating an interrupt request. If an interrupt is to be used to cancel wait mode, that interrupt must first have been enabled and the priority level of the interrupt not to be used for clearing must be set to level 0 before changing to wait mode.

If an interrupt is used to cancel wait mode, the microcomputer selects the clock used when the WAIT instruction is executed for BCLK and restarts operating in that interrupt routine. If only a hardware reset or NMI interrupt will be used to cancel wait mode, the priority level of all interrupts must be set to level 0 before changing to stop mode.

Table 2.16.1 shows the interrupts that can be used for canceling stop mode and wait mode.

(5) BCLK in returning from wait mode or stop mode

(a) Returning from wait mode

The processor immediately returns to the BCLK, which was in use before entering wait mode.

(b) Returning from stop mode

CM06 is set to “1” when the device enters stop mode after selecting the main clock for BCLK. CM17 and CM16 do not change state. In this case, when restored from stop mode, the device starts operating in divided-by-8 mode.

When the device enters stop mode after selecting the subclock for BCLK, CM06, CM17, CM16, and CM07 all do not change state. In this case, when restored from stop mode, the device starts operating in low-speed mode.

Table 2.16.1. Interrupts available for clearing stop mode and wait mode

Interrupt for clearing	Wait mode		Stop mode
	CM02 = 0	CM02 =1(Note 6), CM07 = 0, CM05 = 0	
UART0/UART2 Bus collision detection, Start/stop condition detection interrupt	Possible	Note 1	Note 1
UART1/UART3 Bus collision detection, Start/stop condition detection interrupt	Possible	Note 1	Note 1
DMA0 interrupt	Impossible	Impossible	Impossible
DMA1 interrupt	Impossible	Impossible	Impossible
DMA2 interrupt	Impossible	Impossible	Impossible
DMA3 interrupt	Impossible	Impossible	Impossible
Key input interrupt	Possible	Possible	Possible
A/D interrupt	Note 3	Impossible	Impossible
UART0 transmit/NACK /SSI0 transmit interrupt (Note7)	Possible	Note 1	Note 1
UART0 receive/ACK /SSI0 receive interrupt (Note 7)	Possible	Note 1	Note 1
UART1 transmit/NACK /SSI1 transmit interrupt (Note7)	Possible	Note 1	Note 1
UART1 receive/ACK /SSI1 receive interrupt (Note 7)	Possible	Note 1	Note 1
UART2 transmit/NACK interrupt (Note 7)	Possible	Note 1	Note 1
UART2 receive/ACK interrupt (Note 7)	Possible	Note 1	Note 1
UART3 transmit/NACK interrupt (Note 7)	Possible	Note 1	Note 1
UART3 receive/ACK interrupt (Note 7)	Possible	Note 1	Note 1
Timer A0 interrupt	Possible	Note 2, Note 4	Note 2
Timer A1 interrupt	Possible	Note 2, Note 4	Note 2
Timer A2 interrupt	Possible	Note 2, Note 4	Note 2
Timer A3 interrupt	Possible	Note 2, Note 4	Note 2
Timer A4 interrupt	Possible	Note 2, Note 4	Note 2
USB suspend interrupt	Possible	Possible	Impossible
USB resume interrupt	Possible	Possible	Note 5
USB reset interrupt	Possible	Possible	Impossible
USB SOF interrupt	Possible	Possible	Impossible
USB Vbus detection interrupt	Possible	Possible	Possible
USB function interrupt	Possible	Possible	Impossible
USB EP0 interrupt	Possible	Possible	Impossible
$\overline{\text{INT0}}$ interrupt	Possible	Possible	Possible
$\overline{\text{INT1}}$ interrupt	Possible	Possible	Possible
$\overline{\text{INT2}}$ interrupt	Possible	Possible	Possible
$\overline{\text{NMI}}$ interrupt	Possible	Possible	Possible

Note 1: Can be used when an external clock is selected.
Note 2: Can be used when the external signal is being counted in event counter mode.
Note 3: Can be used in one-shot mode and one-shot sweep mode.
Note 4: Can be used when count source is f_{C32}.
Note 5: Only when USB suspend mode.
Note 6: When the MCU running in low-speed or low power dissipation mode, do not enter wait mode with CM02 is set to "1".
Note 7: When I^2C mode is selected, NACK/ACK, start/stop condition detection interrupt are selected, and when SS pin is selected, trouble error interrupt is selected.

(5) Sequence of returning from stop mode

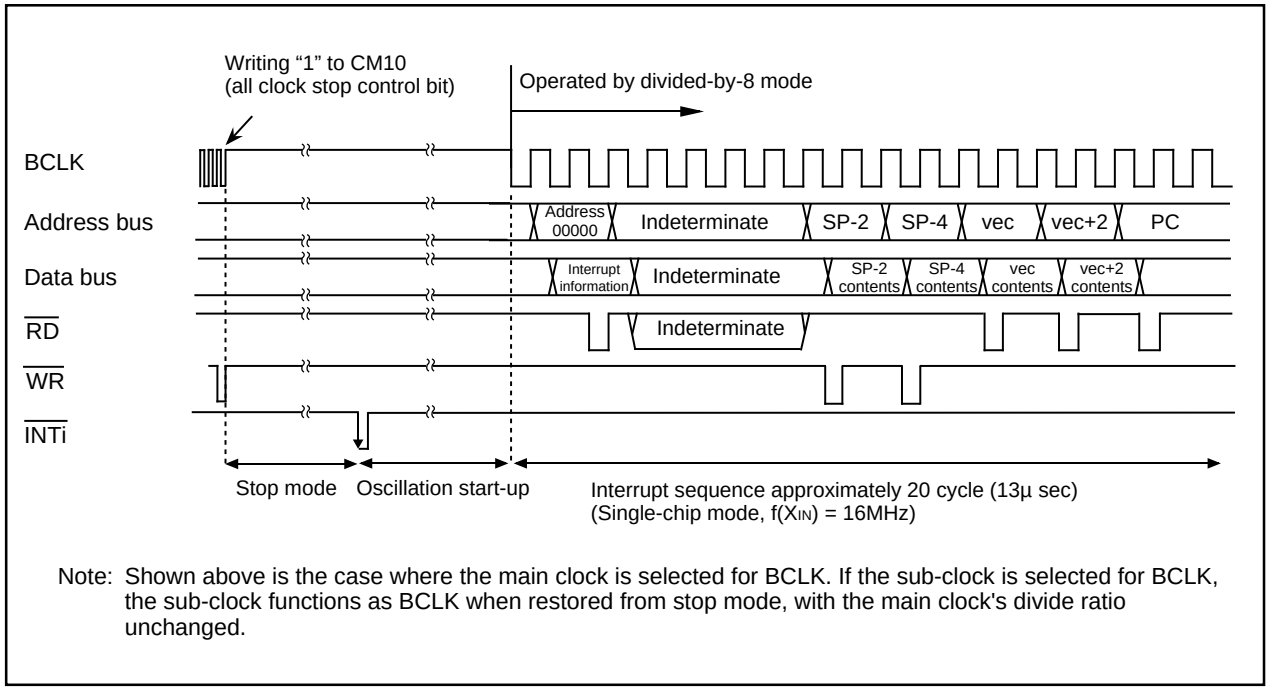
Sequence of returning from stop mode is oscillation start-up time and interrupt sequence.
When interrupt is generated in stop mode, CM10 becomes “0” and clearing stop mode.
Starting oscillation and supplying BCLK execute the interrupt sequence as follow:

In the interrupt sequence, the processor carries out the following in sequence given:

(a) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 0000_{16}. The interrupt request bit of the interrupt written in address 0000_{16} will then be set to “0”.
(b) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
(c) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer assignment flag (U flag) to “0” (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)
(d) Saves the content of the temporary register (Note) within the CPU in the stack area.
(e) Saves the content of the program counter (PC) in the stack area.
(f) Sets the interrupt priority level of the accepted instruction in the IPL.

Note: This register cannot be utilized by the user.
After the interrupt sequence is completed, the processor resumes executing instructions from the first address of the interrupt routine.

Figure 2.16.2 shows the sequence of returning from stop mode.



Note: Shown above is the case where the main clock is selected for BCLK. If the sub-clock is selected for BCLK, the sub-clock functions as BCLK when restored from stop mode, with the main clock's divide ratio unchanged.

Figure 2.16.2. Sequence of returning from stop mode

(6) Registers related to power control

Figure 2.16.3 shows the memory map of power control-related registers, and Figure 2.16.4 shows power control-related registers.

0006_{16}	System clock control register 0 (CM0)
0007_{16}	System clock control register 1 (CM1)

Figure 2.16.3. Memory map of power control-related registers

System clock control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol	Address	When reset
CM0	0006₁₆	48₁₆

Bit Symbol	Bit Name	Function	R	W
Reserved bit		Always set to “0”	O	O
CM02	WAIT peripheral function clock stop bit	0 : Do not stop in wait mode 1 : Stop in wait mode (Note 8)	O	O
CM03	Xcin-Xcout drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM04	Port Xc select bit	0 : I/O port 1 : Xcin-Xcout generation	O	O
CM05	Main clock (Xin-Xout) stop bit (Note 3, 4, 5)	0 : On 1 : Off	O	O
CM06	Main clock division select bit 0 (Note 7)	0 : CM16 and CM17 valid 1 : Divide-by-8 mode	O	O
CM07	System clock select bit (Note 6)	0 : Xin, Xout 1 : Xcin, Xcout	O	O

Note 1: Set bit 0 of the protect register (address 000A16) to “1” before writing to this register.
Note 2: Changes to “1” when changing to stop mode and at a reset.
Note 3: When entering power saving mode, main clock stops using this bit. When returning from stop mode and operating in XIN, set this bit to “0”. When main clock oscillation is operating by itself, set system clock select bit (CM07) to “1” before setting this bit to “1”.
Note 4: When inputting external clock, only clock oscillation buffer is stopped and clock input is acceptable.
Note 5: If this bit is set to “1”, XOUT becomes “H”. The built-in feedback resistor remains connected, so XIN becomes pulled up to Xout (“H”) using the feedback resistor.
Note 6: Set port Xc select bit (CM04) to “1”, and stabilize the sub clock oscillating before setting to this bit from “0” to “1”. Do not write to both bits at the same time. Also, set the main clock stop bit (CM05) to “0”, and stabilize the main clock oscillating before setting this bit from “1” to “0”.
Note 7: This bit changes to “1”, when changing from high-speed/medium mode to stop mode and at reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 8: fC32 is not included. Do not set to “1” when using low-speed or low power dissipation mode.
Note 9: When the XCIN/XCOUT is used, set ports P86 and P87 as the input ports without pull-up.

System clock control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0	0	0	

Symbol	Address	When reset
CM1	0007₁₆	20₁₆

Bit Symbol	Bit Name	Function	R	W
CM10	All clock stop control bit (Note 4)	0 : Clock on 1 : All clocks off (stop mode)	O	O
Reserved bit		Always set to “0”	O	O
CM15	Xin-Xout drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division mode 0 1 : Divide-by-2 mode 1 0 : Divide-by-4 mode 1 1 : Divide-by-16 mode	O	O
CM17			O	O

Note 1: Set bit 0 of the protect register (address 000A16) to “1” before writing to this register.
Note 2: This bit changes to “1”, when changing from high-speed/medium mode to stop mode and at reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 3: Can be selected when bit 6 of the system clock control register 0 (address 0006₁₆) is “0”. If “1”, division mode is fixed at 8.
Note 4: If this bit is set to “1”, XOUT turns “H”, and the built-in feedback resistor is cut off. XCIN and XCOUT turn high-impedance state.

Figure 2.16.4. Power control-related registers

2.16.2 Stop Mode Set-Up

Settings and operation for entering stop mode are described here.

Operation (1) Enables the interrupt used for returning from stop mode.
(2) Sets the interrupt enable flag (I flag) to “1”.
(3) Clearing the protection and setting all clock stop control bit to “1” stops oscillation and causes the processor to go into stop mode.

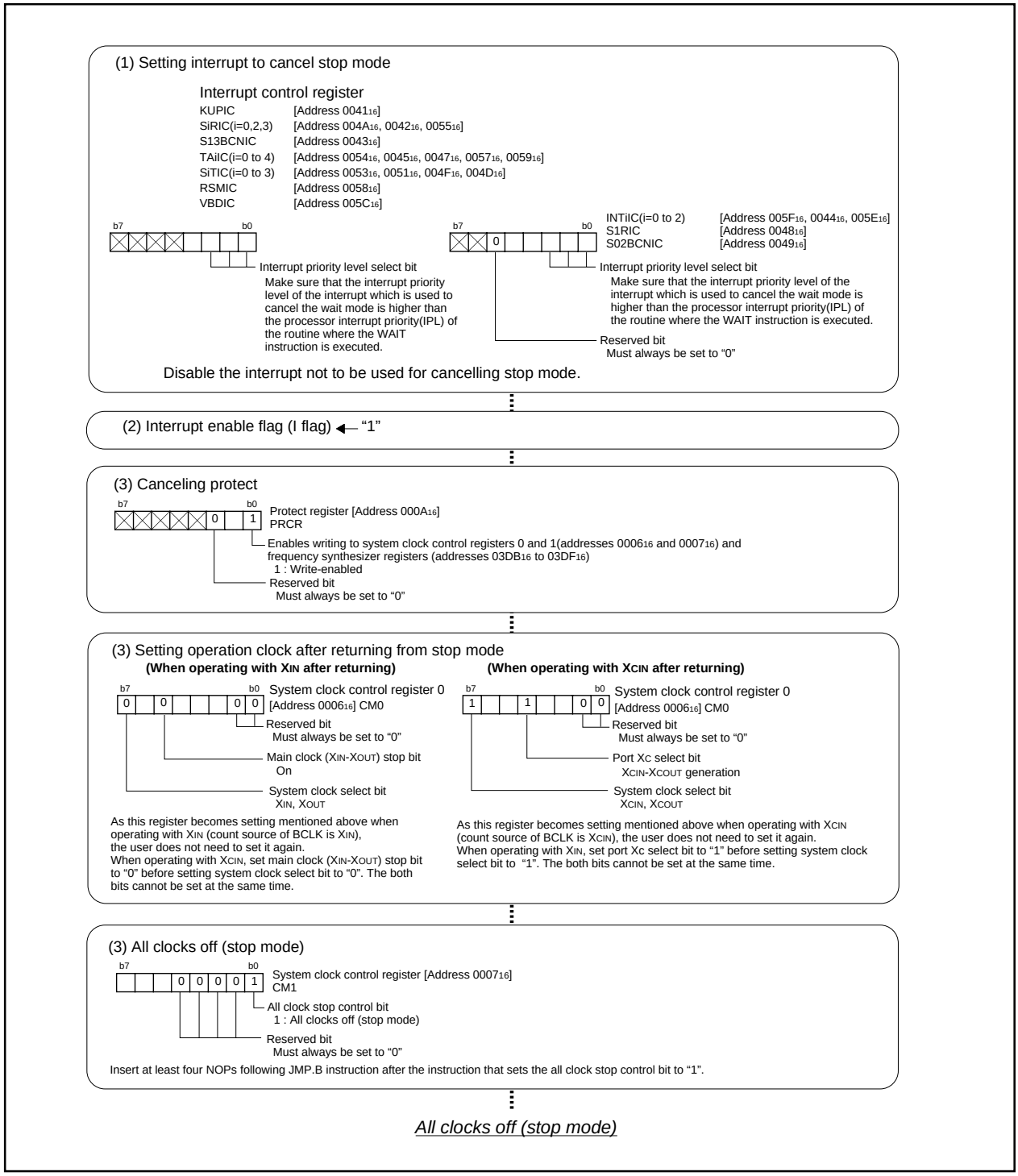

(1) Setting interrupt to cancel stop mode
Interrupt control register
KUPIC [Address 0041₁₆]
SiRIC(i=0,2,3) [Address 004A₁₆, 0042₁₆, 0055₁₆]
S13BCNIC [Address 0043₁₆]
TAiIC(i=0 to 4) [Address 0054₁₆, 0045₁₆, 0047₁₆, 0057₁₆, 0059₁₆]
SiTIC(i=0 to 3) [Address 0053₁₆, 0051₁₆, 004F₁₆, 004D₁₆]
RSMIC [Address 0058₁₆]
VBDIC [Address 005C₁₆]
b7 b0
Interrupt priority level select bit
Make sure that the interrupt priority level of the interrupt which is used to cancel the wait mode is higher than the processor interrupt priority(IPL) of the routine where the WAIT instruction is executed.
INTiIC(i=0 to 2) [Address 005F₁₆, 0044₁₆, 005E₁₆]
S1RIC [Address 0048₁₆]
S02BCNIC [Address 0049₁₆]
b7 b0
0
Interrupt priority level select bit
Make sure that the interrupt priority level of the interrupt which is used to cancel the wait mode is higher than the processor interrupt priority(IPL) of the routine where the WAIT instruction is executed.
Reserved bit
Must always be set to “0”
Disable the interrupt not to be used for cancelling stop mode.
(2) Interrupt enable flag (I flag) ← “1”
(3) Canceling protect
b7 b0
0 1
Protect register [Address 000A₁₆]
PRCR
Enables writing to system clock control registers 0 and 1(addresses 0006₁₆ and 0007₁₆) and frequency synthesizer registers (addresses 03DB₁₆ to 03DF₁₆)
1 : Write-enabled
Reserved bit
Must always be set to “0”
(3) Setting operation clock after returning from stop mode
(When operating with XIN after returning)
b7 b0
0 0 0 0
System clock control register 0
[Address 0006₁₆] CM0
Reserved bit
Must always be set to “0”
Main clock (XIN-XOUT) stop bit
On
System clock select bit
XIN, XOUT
As this register becomes setting mentioned above when operating with XIN (count source of BCLK is XIN), the user does not need to set it again.
When operating with XCIN, set main clock (XIN-XOUT) stop bit to “0” before setting system clock select bit to “0”. The both bits cannot be set at the same time.
(When operating with XCIN after returning)
b7 b0
1 1 0 0
System clock control register 0
[Address 0006₁₆] CM0
Reserved bit
Must always be set to “0”
Port XC select bit
XCIN-XCOUT generation
System clock select bit
XCIN, XCOUT
As this register becomes setting mentioned above when operating with XCIN (count source of BCLK is XCIN), the user does not need to set it again.
When operating with XIN, set port Xc select bit to “1” before setting system clock select bit to “1”. The both bits cannot be set at the same time.
(3) All clocks off (stop mode)
b7 b0
0 0 0 0 1
System clock control register [Address 0007₁₆]
CM1
All clock stop control bit
1 : All clocks off (stop mode)
Reserved bit
Must always be set to “0”
Insert at least four NOPs following JMP.B instruction after the instruction that sets the all clock stop control bit to “1”.
All clocks off (stop mode)

Figure 2.16.5. Example of stop mode set-up

2.16.3 Wait Mode Set-Up

Settings and operation for entering wait mode are described here.

Operation (1) Enables the interrupt used for returning from wait mode.
(2) Sets the interrupt enable flag (I flag) to “1”.
(3) Clears the protection and changes the content of the system clock control register.
(4) Executes the WAIT instruction.

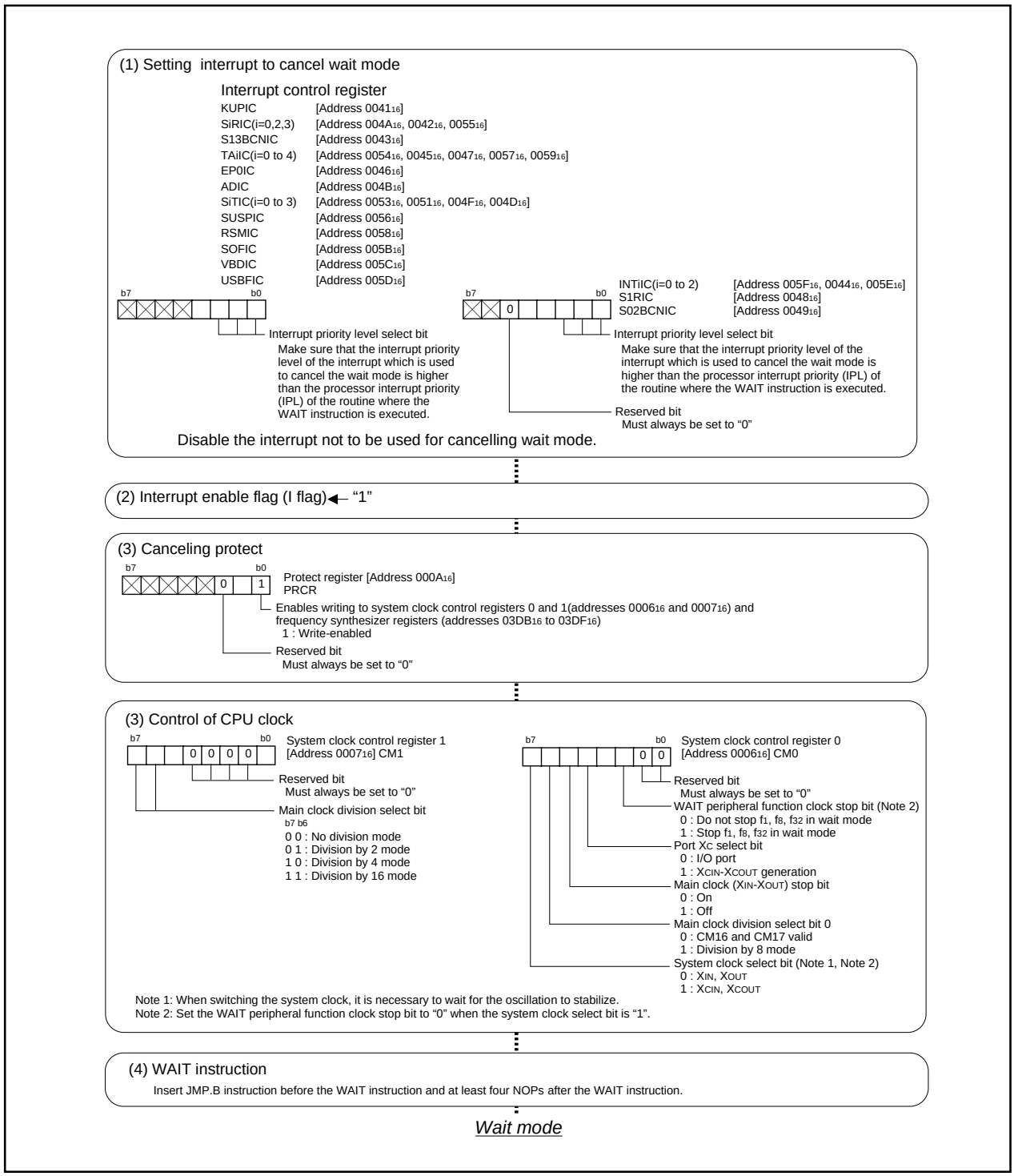

(1) Setting interrupt to cancel wait mode
Interrupt control register
KUPIC [Address 0041₁₆]
SiRIC(i=0,2,3) [Address 004A₁₆, 0042₁₆, 0055₁₆]
S13BCNIC [Address 0043₁₆]
TAiIC(i=0 to 4) [Address 0054₁₆, 0045₁₆, 0047₁₆, 0057₁₆, 0059₁₆]
EP0IC [Address 0046₁₆]
ADIC [Address 004B₁₆]
SiTIC(i=0 to 3) [Address 0053₁₆, 0051₁₆, 004F₁₆, 004D₁₆]
SUSPIC [Address 0056₁₆]
RSMIC [Address 0058₁₆]
SOFIC [Address 005B₁₆]
VBDIC [Address 005C₁₆]
USBFIC [Address 005D₁₆]
INTiIC(i=0 to 2) [Address 005F₁₆, 0044₁₆, 005E₁₆]
S1RIC [Address 0048₁₆]
S02BCNIC [Address 0049₁₆]
b7 b0
Interrupt priority level select bit
Make sure that the interrupt priority level of the interrupt which is used to cancel the wait mode is higher than the processor interrupt priority (IPL) of the routine where the WAIT instruction is executed.
Interrupt priority level select bit
Make sure that the interrupt priority level of the interrupt which is used to cancel the wait mode is higher than the processor interrupt priority (IPL) of the routine where the WAIT instruction is executed.
Reserved bit
Must always be set to “0”
Disable the interrupt not to be used for cancelling wait mode.
(2) Interrupt enable flag (I flag)◄— “1”
(3) Canceling protect
Protect register [Address 000A₁₆]
PRCR
Enables writing to system clock control registers 0 and 1(addresses 0006₁₆ and 0007₁₆) and frequency synthesizer registers (addresses 03DB₁₆ to 03DF₁₆)
1 : Write-enabled
Reserved bit
Must always be set to “0”
(3) Control of CPU clock
System clock control register 1
[Address 0007₁₆] CM1
Reserved bit
Must always be set to “0”
Main clock division select bit
b7 b6
0 0 : No division mode
0 1 : Division by 2 mode
1 0 : Division by 4 mode
1 1 : Division by 16 mode
System clock control register 0
[Address 0006₁₆] CM0
Reserved bit
Must always be set to “0”
WAIT peripheral function clock stop bit (Note 2)
0 : Do not stop f1, f8, f32 in wait mode
1 : Stop f1, f8, f32 in wait mode
Port Xc select bit
0 : I/O port
1 : XCIN-XCOUT generation
Main clock (XIN-XOUT) stop bit
0 : On
1 : Off
Main clock division select bit 0
0 : CM16 and CM17 valid
1 : Division by 8 mode
System clock select bit (Note 1, Note 2)
0 : XIN, XOUT
1 : XCIN, XCOUT
Note 1: When switching the system clock, it is necessary to wait for the oscillation to stabilize.
Note 2: Set the WAIT peripheral function clock stop bit to “0” when the system clock select bit is “1”.
(4) WAIT instruction
Insert JMP.B instruction before the WAIT instruction and at least four NOPs after the WAIT instruction.
Wait mode

Figure 2.16.6. Example of wait mode set-up

2.16.4 Precautions in Power Control

(1) The processor does not switch to stop mode when the $\overline{\text{NMI}}$ pin is at “L” level.

(2) When returning from stop mode by hardware reset, $\overline{\text{RESET}}$ pin must be set to “L” level until main clock oscillation is stabilized.

(3) When entering wait mode, insert a JMP.B instruction before a WAIT instruction. Do not execute any instructions which can generate a write to RAM between the JMP.B and WAIT instructions. Disable the DMA transfers, if a DMA transfer may occur between the JMP.B and WAIT instructions. After the WAIT instruction, insert at least 4 NOP instructions. When entering wait mode, the instruction queue roadstead the instructions following WAIT, and depending on timing, some of these may execute before the microcomputer enters wait mode.

Program example when entering wait mode

```
Program Example:    JMP.B   L1      ; Insert JMP.B instruction before WAIT instruction
                L1:
                    FSET    I       ;
                    WAIT            ; Enter wait mode
                    NOP             ; More than 4 NOP instructions
                    NOP
                    NOP
                    NOP
```

(4) When entering stop mode, insert a JMP.B instruction immediately after executing an instruction which sets the CM10 bit in the CM1 register to “1”, and then insert at least 4 NOP instructions. When entering stop mode, the instruction queue reads ahead the instructions following the instruction which sets the CM10 bit to “1” (all clock stops), and, some of these may execute before the microcomputer enters stop mode or before the interrupt routine for returning from stop mode.

Program example when entering stop mode

```
Program Example:    FSET    I
                    BSET    CM10        ; Enter stop mode
                    JMP.B   L2          ; Insert JMP.B instruction
                L2:
                    NOP                 ; More than 4 NOP instructions
                    NOP
                    NOP
                    NOP
```

(5) Before the count source for BCLK can be changed from XIN to XCIN or vice versa, the clock to which the count source is going to be switched must be oscillating stably. Allow a wait time in software for the oscillation to stabilize before switching over the clock.

(6) Suggestions to reduce power consumption

(a) Ports

The processor retains the state of each programmable I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A pass current flows in input ports that float. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

(b) Memory expansion mode and microprocessor mode

When the MCU enters wait mode while operating in memory expansion mode or microprocessor mode, a pin functioning as part of the address or data bus retains it's state on the bus before wait mode is entered. Shift to single-chip mode and output an arbitrary value in order to reduce current consumption. By shifting to single-chip mode, a pin which was functioning as part of the bus becomes a general-purpose port and can output an arbitrary value. Set the port registers and direction registers after shifting to single-chip mode (this implies that any control pins ($\overline{CS}$, $\overline{WR}$, $\overline{RD}$, etc..) being used for access of an external device be changed as well).

The same applies to stop mode.

(c) A/D converter

A current always flows in the VREF pin. When entering wait mode or stop mode, set the Vref connection bit to “0” so that no current flows into the VREF pin.

(d) Stopping peripheral functions

In wait mode, stop non-used peripheral functions using the WAIT peripheral function clock stop bit. However, peripheral function clock f_{C32} does not stop so that the peripherals using f_{C32} do not contribute to the power saving. When the MCU running in low-speed or low power dissipation mode, do not enter WAIT mode with this bit set to “1”.

(e) External clock

When using an external clock input for the CPU clock, set the main clock stop bit to “1”. Setting the main clock stop bit to “1” causes the XOUT pin not to operate and the power consumption goes down (when using an external clock input, the clock signal is input regardless of the content of the main clock stop bit).

2.17 Programmable I/O Ports Usage

2.17.1 Overview of the programmable I/O ports usage

Eighty-one programmable I/O ports and one input-only port are available. I/O pins also serve as I/O pins for built-in peripheral functions.

Each port has a direction register that defines the I/O direction and also has a port register for I/O data. In addition, each port has a pull-up control register that defines pull-up in terms of 4 bits. The input-only port has neither direction register nor pull-up control bit.

The following is an overview of the programmable I/O ports usage:

(1) Writing to a port register

With the direction register set to output, the level of the written values from each relevant pin is output by writing to a port register. The output level conforms to CMOS output. Port $P7_0$ and $P7_1$ are N channel open drain. Writing to the port register, with the direction register set to input, inputs a value to the port register, but nothing is output to the relevant pins. The output level remains floating.

In memory expansion and microprocessor mode, the contents of corresponding port and direction registers of pins A0 to A19, D0 to D15, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK cannot be modified.

(2) Reading a port register

With the direction register set to output, reading a port register takes out the content of the port register, not the content of the pin. With the direction register set to input, reading the port register takes out the content of the pin.

(3) Input-only port

$P8_5$ is used as the input-only port, it also serves as $\overline{NMI}$. $P8_5$ has no direction register. Pull-up cannot be set to this port. As $\overline{NMI}$ cannot be disabled, an $\overline{NMI}$ interrupt occurs if a falling edge is input to $P8_5$. Use $P8_5$ for reading the level input at this time only.

(4) Setting pull-up

The pull-up control bit allows setting of the pull-up, in terms of 4 bits, either in use or not in use. For the four bits chosen, pull-up is effective only in the ports whose direction register is set to input. Pull-up is not effective in ports whose direction register is set to output.

Do not set pull-up of corresponding pin when X_{CIN}/X_{COUT} is set or a port is used as A/D input.

Pull-up can be set for P0 to P3, $P4_0$ to $P4_3$, $P5_0$ to $P5_3$ in only single-chip mode. Pull-up cannot be set for P0 to P3, $P4_0$ to $P4_3$, $P5_0$ to $P5_3$ in memory expansion and microprocessor modes. The contents of register can be changed, but the pull-up resistance is not connected.

(5) Setting drive capacity

A normal drive and N-channel high drive for the N-channel transistor drive capacity of port P7 can be selected. Port P7 can be configured to drive an LED by increasing the drive strength of the corresponding N-channel transistor bits.

(6) I/O functions of built-in peripheral devices

Table 2.17.1 shows relation between ports and I/O functions of built-in peripheral devices.

Table 2.17.1. Relation between ports and I/O functions of built-in peripheral devices

Port	Internal peripheral device I/O pins
P6	I/O pins/Serial sound interface, I^2C and SPI communication pins for UART0 and UART1
P7	Timer A0 to A3 I/O pins / I/O pins or I^2C, SPI communication pins for UART2 and UART3 / LED drive output pins
$P8_0$, $P8_1$	Timer A4 I/O pins
$P8_2$, $P8_3$, $P8_4$	Input pins for external interrupt
$P8_6$, $P8_7$	Sub-clock oscillation circuit I/O pins
$P9_0$	Attach/Detach control pin for USB
$P9_2$	$\overline{SOF}$ output pin for USB
$P9_3$	A/D trigger input pin
P10	A/D converter input pins / key-input interrupt function input pins

(7) Examples of working on non-used pins

Table 2.17.2 contains examples of working on non-used pins. There are shown here for mere examples. In practical use, make suitable changes and perform sufficient evaluation in compliance with you application.

(a) Single-chip mode

Table 2.17.2. Examples of working on unused pins in single-chip mode

Pin name	Connection
Ports P0 to P10 (excluding $P8_5$)	After setting for input mode, connect every pin to V_{SS} or V_{CC} via a resistor; or after setting for output mode, leave these pins open. (Note 1, Note 2, Note 3)
X_{OUT} (Note 1)	Open
$\overline{NMI}$	Connect to V_{CC} via a resistor (pull-up)
UV_{CC}, AV_{CC}	Connect to V_{CC}
AV_{SS}, V_{REF}, BYTE	Connect to V_{SS}
USB D+, USB D-	Open
LPF	Open
VbusDTCT	Open

Note 1: When an external clock is input to the X_{IN} pin.
Note 2: If setting these pins in output mode and opening them, ports are in input mode until switched into output mode by use of software after reset. Thus the voltage levels of the pins become unstable, and there can be instances in which the power source current increases while the ports are in input mode.
In view of an instance in which the contents of the direction registers change due to a runaway generated by noise or other causes, setting the contents of the direction registers periodically by use of software increases program reliability.
Note 3: Output “L” if port $P7_0$ and $P7_1$ are set to output mode.
Port $P7_0$ and $P7_1$ are N channel open drain output.

(b) Memory expansion mode, microprocessor mode

Table 2.17.3. Examples of working on unused pins in memory expansion mode or microprocessor mode

Pin name	Connection
Ports P6 to P10 (excluding P85)	After setting for input mode, connect every pin to VSS or VCC via a resistor; or after setting for output mode, leave these pins open. (Note 2, Note 3, Note 4)
P45/$\overline{CS1}$ to P47/$\overline{CS3}$	After setting for port input mode and setting CS1 to CS3 output enable bit to “0”, connect to VCC via a resistor (pull-up)
$\overline{BHE}$(Note 5), ALE(Note 5), $\overline{HLDA}$(Note 5), XOUT (Note 1), BCLK	Open
$\overline{HOLD}$, $\overline{RDY}$, $\overline{NMI}$	Connect via resistor to VCC (pull-up)
UVCC, AVCC	Connect to VSS
AVSS, VREF	Open
USB D+, USB D-	Open
LPF	Open
VbusDTCT (Note 6)	Open

Note 1: When an external clock is input to the XIN pin.
Note 2: If setting these pins in output mode and opening them, ports are in input mode until switched into output mode by use of software after reset. Thus the voltage levels of the pins become unstable, and there can be instances in which the power source current increases while the ports are in input mode.
In view of an instance in which the contents of the direction registers change due to a runaway generated by noise or other causes, setting the contents of the direction registers periodically by use of software increases program reliability.
Note 3: Make wiring as short as possible (not more than 2 cm from the microcomputer's pins) in working on non-used pins.
Note 4: Output “L” if port P70 and P71 are set to output mode.
Port P70 and P71 are N channel open drain output.
Note 5: When a VSS level is connected to the CNVSS pin, these pins are input ports until the processor mode is switched by use of software after reset. Thus the voltage levels of the pins destabilize, and there can be an increase in the power source current while these pins are input ports.
Note 6: VbusDTCT pin is pulled down internaly.

(8) Registers related to the programmable I/O ports

Figure 2.17.1 shows the memory map of programmable I/O ports-related registers, and Figures 2.17.2 to 2.17.5 show programmable I/O ports-related registers.

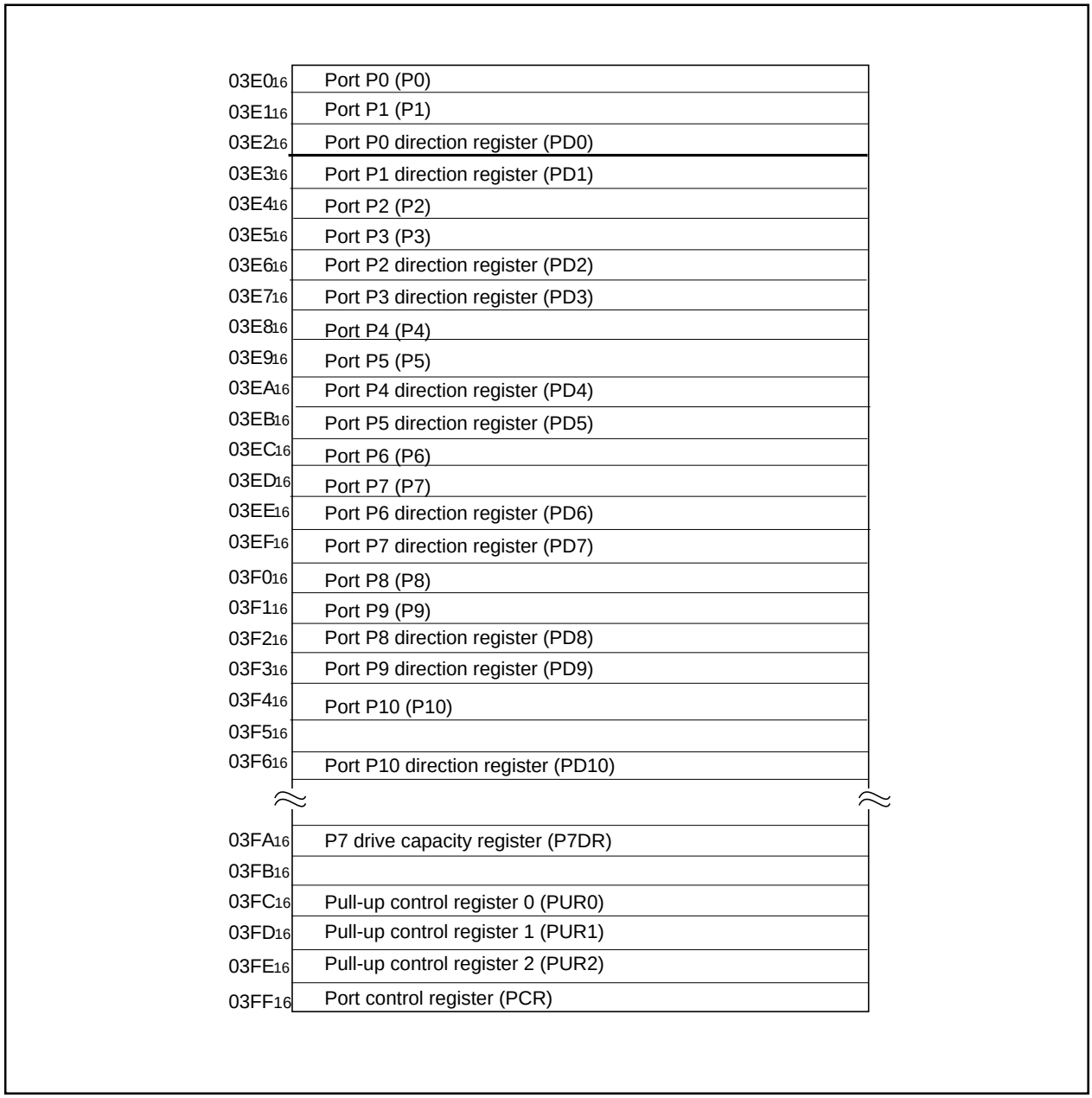

03E016 Port P0 (P0)
03E116 Port P1 (P1)
03E216 Port P0 direction register (PD0)
03E316 Port P1 direction register (PD1)
03E416 Port P2 (P2)
03E516 Port P3 (P3)
03E616 Port P2 direction register (PD2)
03E716 Port P3 direction register (PD3)
03E816 Port P4 (P4)
03E916 Port P5 (P5)
03EA16 Port P4 direction register (PD4)
03EB16 Port P5 direction register (PD5)
03EC16 Port P6 (P6)
03ED16 Port P7 (P7)
03EE16 Port P6 direction register (PD6)
03EF16 Port P7 direction register (PD7)
03F016 Port P8 (P8)
03F116 Port P9 (P9)
03F216 Port P8 direction register (PD8)
03F316 Port P9 direction register (PD9)
03F416 Port P10 (P10)
03F516
03F616 Port P10 direction register (PD10)
03FA16 P7 drive capacity register (P7DR)
03FB16
03FC16 Pull-up control register 0 (PUR0)
03FD16 Pull-up control register 1 (PUR1)
03FE16 Pull-up control register 2 (PUR2)
03FF16 Port control register (PCR)

Figure 2.17.1. Memory map of programmable I/O ports-related registers

Port 7 drive capacity register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: P7DR
Address: 03FA16
When reset: 0016

Bit symbol	Bit name	Function	R	W
P7DR0	P70 LED drive capacity	The N-channel high drive capacity is activated for the corresponding bit. 0 : Normal drive 1 : N-channel high drive	O	O
P7DR1	P71 LED drive capacity		O	O
P7DR2	P72 LED drive capacity		O	O
P7DR3	P73 LED drive capacity		O	O
P7DR4	P74 LED drive capacity		O	O
P7DR5	P75 LED drive capacity		O	O
P7DR6	P76 LED drive capacity		O	O
P7DR7	P77 LED drive capacity		O	O

Port control register

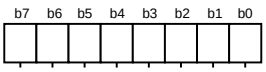
b7	b6	b5	b4	b3	b2	b1	b0

Symbol: PCR
Address: 03FF16
When reset: 0016

Bit Symbol	Bit Name	Function	R	W
PCR0	Port P1control register	0 : When input port, read port input level. When output port, read the contents of Port P1 register. 1 : Read the contents of Port P1 register through input/output port.	O	O
OECTRL	AND Flash OE control bit	0 : Data read mode enabled 1 : Output disabled	O	O
WECTRL	AND Flash WE control bit	0 : Input disabled 1 : Command/Address mode enabled	O	O
AFPE	AND Flash port enable bit	0 : P0 & P1(0-2) GPI/O function 1 : P0 & P1(0-2) AND Flash control function	O	O
Nothing is assigned. Write “0” when writing to this bit. The value is “0” when read.			–	–

Figure 2.17.2. Programmable I/O ports-related registers (1)

Port Pi direction register (Note 1)

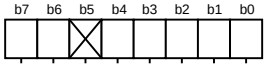


Symbol	Address	When reset
PDi (i = 0 to 7, 10)	03E2$_{16}$, 03E3$_{16}$, 03E6$_{16}$, 03E7$_{16}$, 03EA$_{16}$ 03EB$_{16}$, 03EE$_{16}$, 03EF$_{16}$, 03F6$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 7, 10)	O	O
PDi_1	Port Pi$_1$ direction register		O	O
PDi_2	Port Pi$_2$ direction register		O	O
PDi_3	Port Pi$_3$ direction register		O	O
PDi_4	Port Pi$_4$ direction register		O	O
PDi_5	Port Pi$_5$ direction register		O	O
PDi_6	Port Pi$_6$ direction register		O	O
PDi_7	Port Pi$_7$ direction register		O	O

Note 1: In memory expansion and microprocessor mode, the contents of corresponding port Pi direction register of pins A_0 to A_{19}, D_0 to D_{15}, $\overline{CS_0}$ to $\overline{CS_3}$, $\overline{RD}$, $\overline{WRL}/\overline{WR}$, $\overline{WRH}/\overline{BHE}$, ALE, RDY, HOLD, HLDA and BCLK cannot be modified.

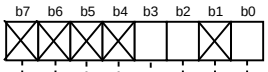
Port P8 direction register



Symbol	Address	When reset
PD8	03F2$_{16}$	00X00000$_2$

Bit symbol	Bit name	Function	R	W
PD8_0	Port P8$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	O	O
PD8_1	Port P8$_1$ direction register		O	O
PD8_2	Port P8$_2$ direction register		O	O
PD8_3	Port P8$_3$ direction register		O	O
PD8_4	Port P8$_4$ direction register		O	O
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.			–	–
PD8_6	Port P8$_6$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	O	O
PD8_7	Port P8$_7$ direction register		O	O

Port P9 direction register



Symbol	Address	When reset
PD9	03F3$_{16}$	XXXX00X0$_2$

Bit symbol	Bit name	Function	R	W
PD9_0	Port P9$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	O	O
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.			–	–
PD9_2	Port P9$_2$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	O	O
PD9_3	Port P9$_3$ direction register		O	O
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.			–	–

Figure 2.17.3. Programmable I/O ports-related registers (2)

Port Pi register (Note 2)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: Pi (i = 0 to 7, 10)

Address: 03E0$_{16}$, 03E1$_{16}$, 03E4$_{16}$, 03E5$_{16}$, 03E8$_{16}$, 03E9$_{16}$, 03EC$_{16}$, 03ED$_{16}$, 03F4$_{16}$

When reset: Indeterminate / Indeterminate

Bit symbol	Bit name	Function	R	W
Pi_0	Port Pi$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : "L" level data 1 : "H" level data (Note 1) (i = 0 to 7, 10)	○	○
Pi_1	Port Pi$_1$ register		○	○
Pi_2	Port Pi$_2$ register		○	○
Pi_3	Port Pi$_3$ register		○	○
Pi_4	Port Pi$_4$ register		○	○
Pi_5	Port Pi$_5$ register		○	○
Pi_6	Port Pi$_6$ register		○	○
Pi_7	Port Pi$_7$ register		○	○

Note 1: Since P7$_0$ and P7$_1$ are N-channel open drain ports, the data is high-impedance.
Note 2: In memory expansion and microprocessor mode, the contents of corresponding port Pi register of pins A$_0$ to A$_{19}$, D$_0$ to D$_{15}$, $\overline{CS_0}$ to $\overline{CS_3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK cannot be modified.

Port P8 register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: P8

Address: 03F0$_{16}$

When reset: Indeterminate

Bit symbol	Bit name	Function	R	W
P8_0	Port P8$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit (except for P8$_5$) 0 : "L" level data 1 : "H" level data	○	○
P8_1	Port P8$_1$ register		○	○
P8_2	Port P8$_2$ register		○	○
P8_3	Port P8$_3$ register		○	○
P8_4	Port P8$_4$ register		○	○
P8_5	Port P8$_5$ register		○	×
P8_6	Port P8$_6$ register		○	○
P8_7	Port P8$_7$ register		○	○

Port P9 register

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	☒				

Symbol: P9

Address: 03F1$_{16}$

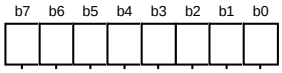
When reset: Indeterminate

Bit symbol	Bit name	Function	R	W
P9_0	Port P9$_0$ register	0 : "L" level data 1 : "H" level data	○	○
VbusDTCT	Vbus detect state bit	0 : Not powered 1 : Powered (Note 1)	○	×
P9_2	Port P9$_2$ register	0 : "L" level data 1 : "H" level data	○	○
P9_3	Port P9$_3$ register	0 : "L" level data 1 : "H" level data	○	○
Nothing is assigned. Write "0" when writing to these bits. The value is indeterminate if read.			–	–

Note 1: This pin cannot be used for GPI/O. This bit reads "0" when Vbus detect is disabled.

Figure 2.17.4. Programmable I/O ports-related registers (3)

Pull-up control register 0 (Note)

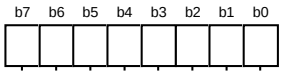


Symbol	Address	When reset
PUR0	03FC16	0016

Bit symbol	Bit name	Function	R	W
PU00	P00 to P03 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU01	P04 to P07 pull-up		○	○
PU02	P10 to P13 pull-up		○	○
PU03	P14 to P17 pull-up		○	○
PU04	P20 to P23 pull-up		○	○
PU05	P24 to P27 pull-up		○	○
PU06	P30 to P33 pull-up		○	○
PU07	P34 to P37 pull-up		○	○

Note : In memory expansion and microprocessor mode, the content of this register can be changed, but the pull-up resistance is not connected.

Pull-up control register 1

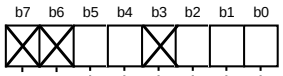


Symbol	Address	When reset
PUR1	03FD16	0016 (Note 2)

Bit symbol	Bit name	Function	R	W
PU10	P40 to P43 pull-up (Note 3)	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU11	P44 to P47 pull-up		○	○
PU12	P50 to P53 pull-up (Note 3)		○	○
PU13	P54 to P57 pull-up		○	○
PU14	P60 to P63 pull-up		○	○
PU15	P64 to P67 pull-up		○	○
PU16	P72 to P73 pull-up (Note 1)		○	○
PU17	P74 to P77 pull-up		○	○

Note 1: Since P70 and P71 are N-channel open drain ports, pull-up is not available for them.
Note 2: This register becomes 0216 when reset under the following conditions:
a) Hardware reset: when VCC is applied to the CNVSS pin.
b) Software reset: if bit 1 and bit 0 of processor mode register 0 (address 000416) [102] or[112] before reset.
Note 3: In memory expansion and microprocessor mode, the content of these bits can be changed, but the pull-up resistance is not connected.

Pull-up control register 2



Symbol	Address	When reset
PUR2	03FE16	0016

Bit symbol	Bit name	Function	R	W
PU20	P80 to P83 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU21	P84 to P87 pull-up (Except P85)		○	○
PU22	P90 to P93 pull-up		○	○
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be “0”.			—	—
PU24	P100 to P103 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU25	P104 to P107 pull-up		○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			—	—

Figure 2.17.5. Programmable I/O ports-related registers (4)

THIS PAGE IS BLANK FOR REASONS OF LAYOUT.

Chapter 3

Examples of Peripheral Functions Applications

This chapter presents applications in which peripheral functions built in the M30245 are used. They are shown here as examples. In practical use, make suitable changes and perform sufficient evaluation. For basic use, see Chapter 2 Peripheral Functions Usage.

3.1 Long-Period Timers

Overview In this process, Timer A0 and Timer A1 are connected to make a 16-bit timer with a 16-bit prescaler. Figure 3.1.1 shows the operation timing, Figure 3.1.2 shows the connection diagram, and Figures 3.1.3 and 3.1.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- Event counter mode of timer A

Specifications

(1) Set timer A0 to timer mode, and set timer A1 to event counter mode.

(2) Perform a count on count source f_1 using timer A0 to count for 1 ms, and perform a count on timer A0 using timer A1 to count for 1 second.

(3) Connect a 16-MHz oscillator to XIN.

Operation (1) Setting the count start flag to “1” causes the counter to begin counting. The counter of timer A0 performs a down count on count source f_1.

(2) If the counter of timer A0 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer A0 interrupt request bit goes to “1”. The counter of timer A1 performs a down count on underflows in timer A0.

(3) If the counter of timer A1 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer A1 interrupt request bit goes to “1”.

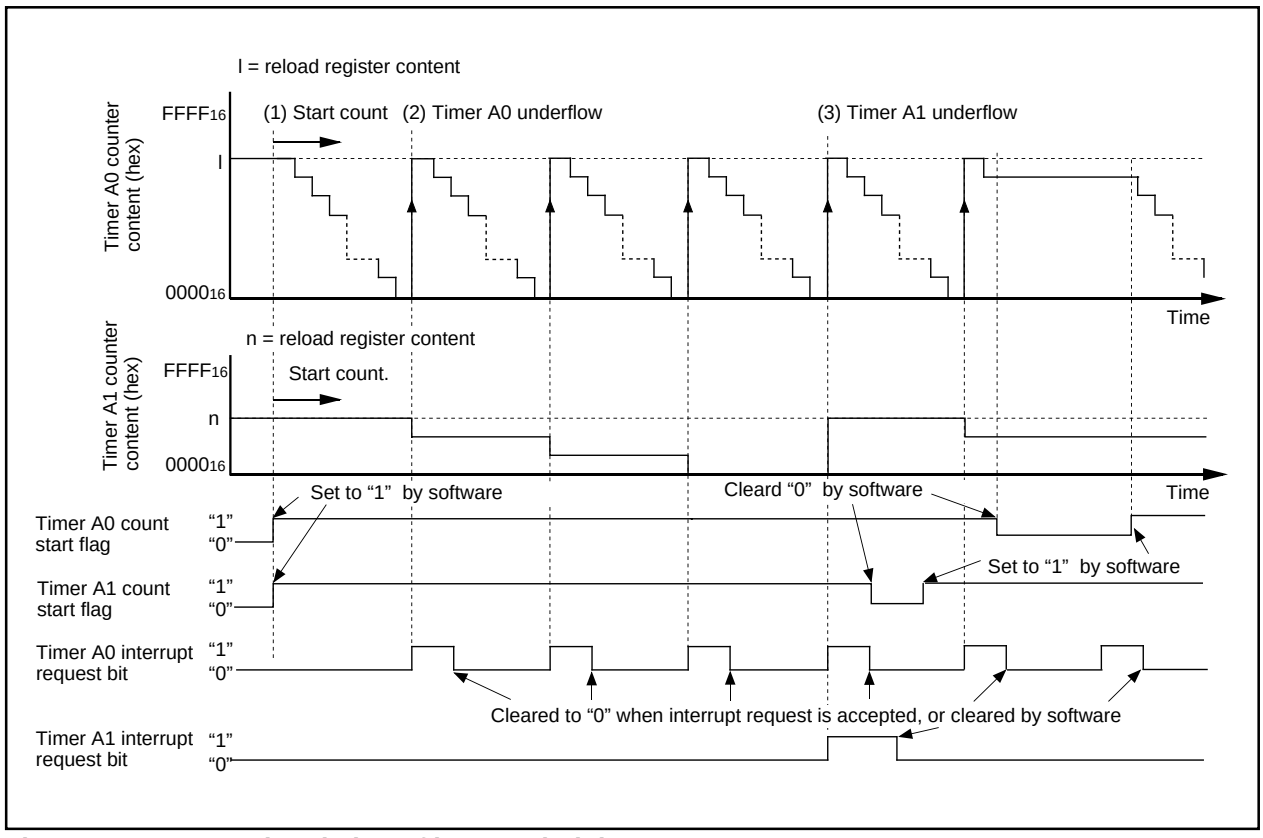


Figure 3.1.1. Operation timing of long-period timers

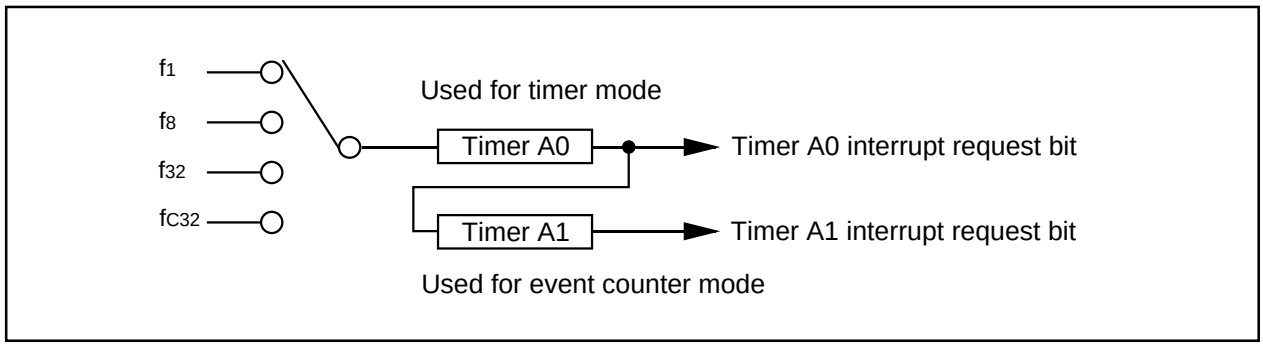


Figure 3.1.2. Connection diagram of long-period timers

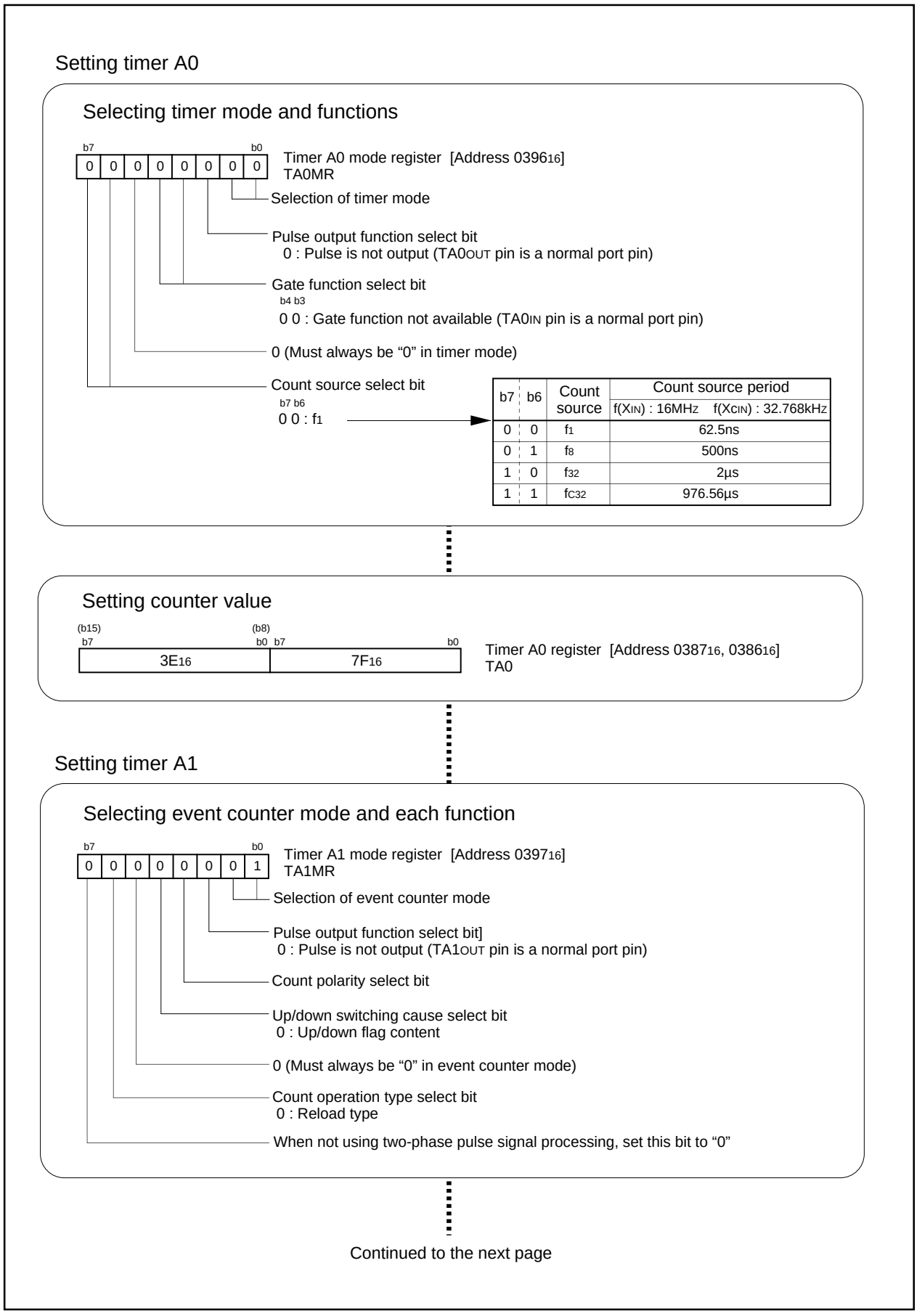

Setting timer A0
Selecting timer mode and functions
b7 b0
0 0 0 0 0 0 0 0
Timer A0 mode register [Address 039616]
TA0MR
Selection of timer mode
Pulse output function select bit
0 : Pulse is not output (TA0OUT pin is a normal port pin)
Gate function select bit
b4 b3
0 0 : Gate function not available (TA0IN pin is a normal port pin)
0 (Must always be "0" in timer mode)
Count source select bit
b7 b6
0 0 : f1
b7 b6 Count source Count source period
f(XIN) : 16MHz f(XCIN) : 32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fC32 976.56µs
Setting counter value
(b15) (b8)
b7 b0 b7 b0
3E16 7F16
Timer A0 register [Address 038716, 038616]
TA0
Setting timer A1
Selecting event counter mode and each function
b7 b0
0 0 0 0 0 0 0 1
Timer A1 mode register [Address 039716]
TA1MR
Selection of event counter mode
Pulse output function select bit]
0 : Pulse is not output (TA1OUT pin is a normal port pin)
Count polarity select bit
Up/down switching cause select bit
0 : Up/down flag content
0 (Must always be "0" in event counter mode)
Count operation type select bit
0 : Reload type
When not using two-phase pulse signal processing, set this bit to "0"
Continued to the next page

Figure 3.1.3. Set-up procedure of long-period timers (1)

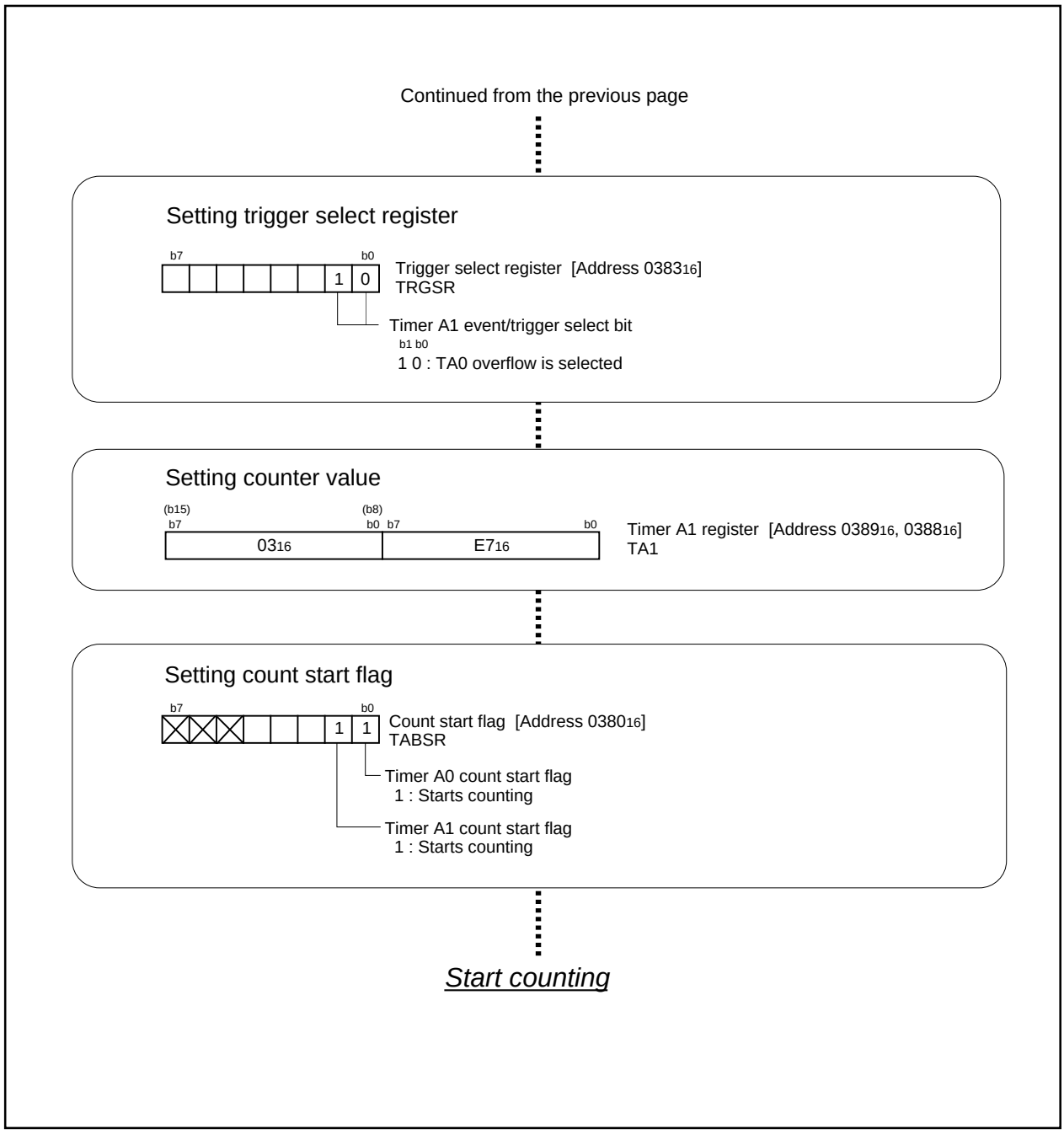


Figure 3.1.4. Set-up procedure of long-period timers (2)

3.2 Variable-Period Variable-Duty PWM Output

Overview In this process, Timer A0 and A1 are used to generate variable-period, variable-duty PWM output. Figure 3.2.1 shows the operation timing, Figure 3.2.2 shows the connection diagram, and Figures 3.2.3 and 3.2.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- One-shot timer mode of timer A

Specifications

(1) Set timer A0 in timer mode, and set timer A1 in one-shot timer mode with pulse-output function.

(2) Set 1 ms, the PWM period, to timer A0. Set 500 µs, the width of PWM "H" pulse, to timer A1. Both timer A0 and timer A1 use f_1 for the count source.

(3) Connect a 16-MHz oscillator to XIN.

Operation (1) Setting the count start flag to "1" causes the counter of timer A0 to begin counting. The counter of timer A0 performs a down count on count source f_1.

(2) If the counter of timer A0 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer A0 interrupt request bit goes to "1".

(3) An underflow in timer A0 triggers the counter of timer A1 and causes it to begin counting. When the counter of timer A1 begins counting, the output level of the TA1OUT pin goes to "H".

(4) As soon as the count of the counter of timer A1 becomes "0000_{16}", the output level of TA1OUT pin goes to "L", and the counter reloads the content of the reload register and stops counting. At the same time, the timer A1 interrupt request bit goes to "1".

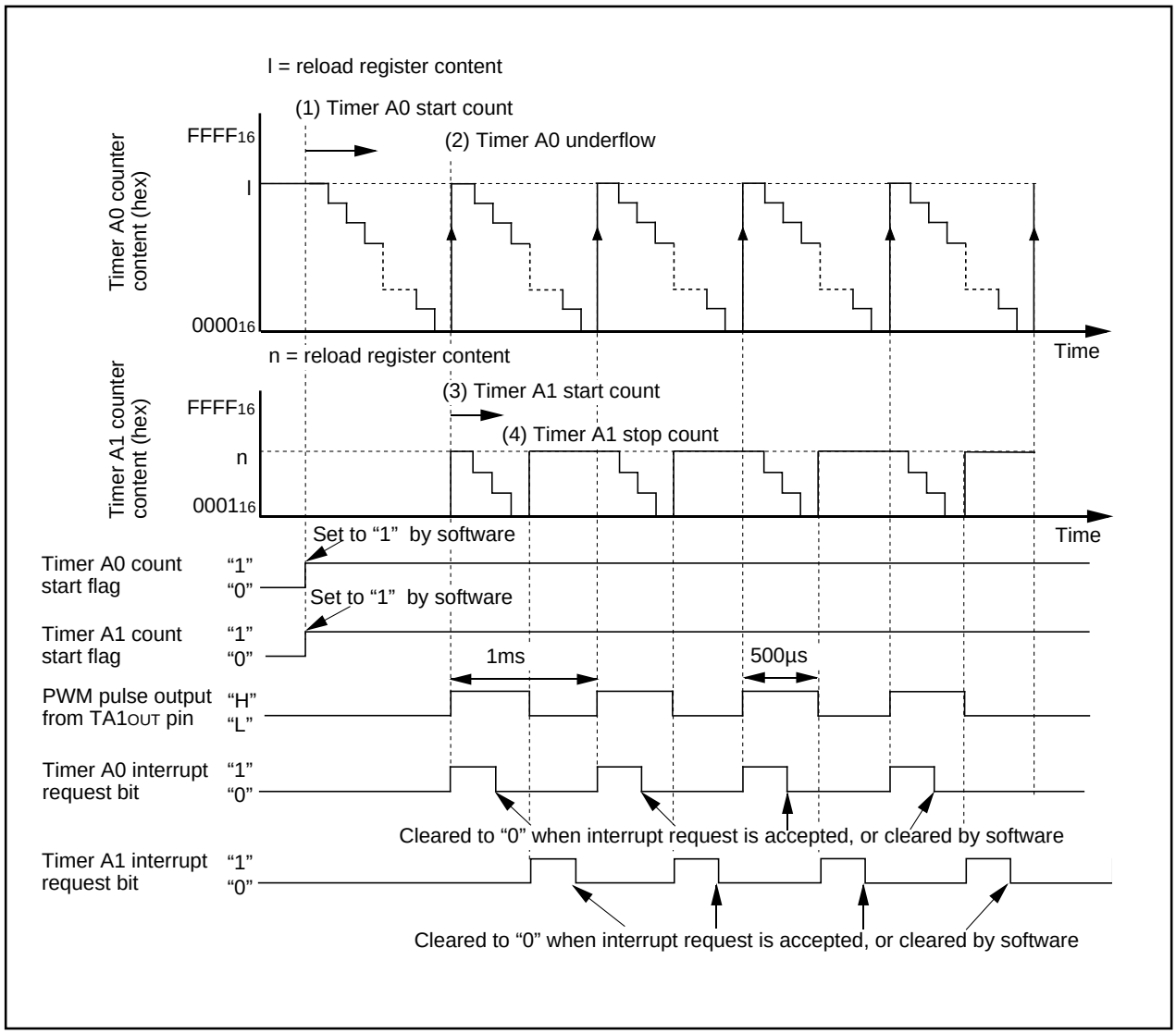


Figure 3.2.1. Operation timing of variable-period variable-duty PWM output

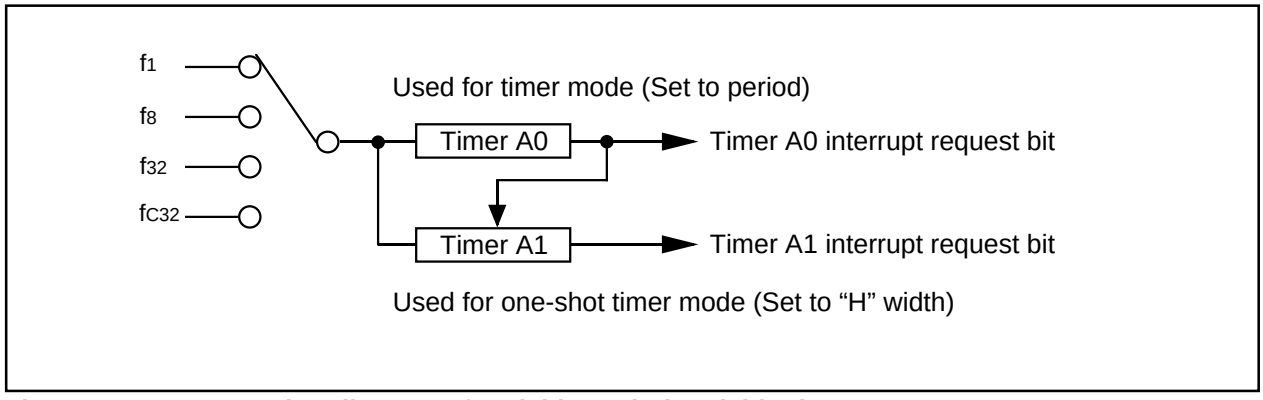


Figure 3.2.2. Connection diagram of variable-period variable-duty PWM output

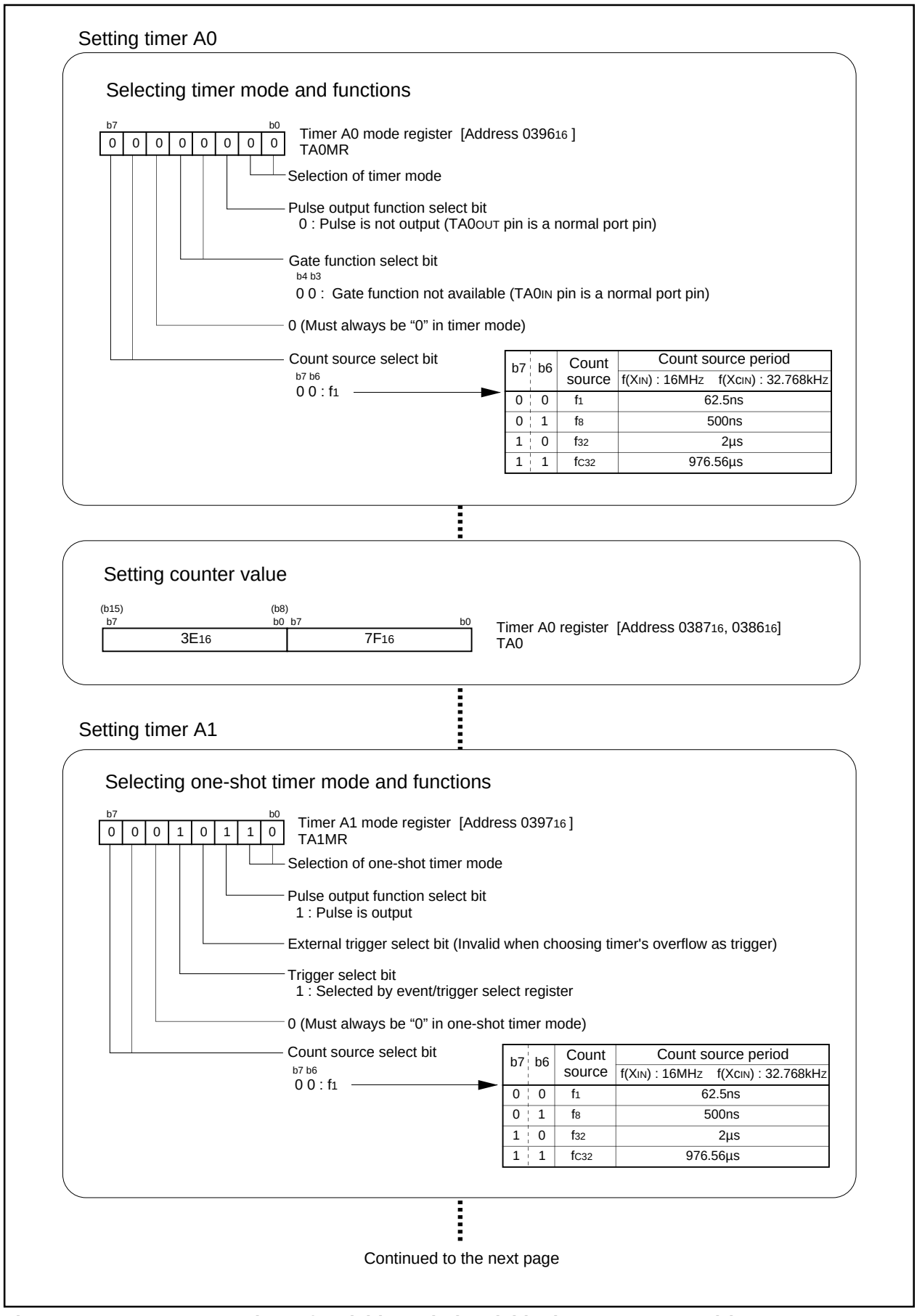

Setting timer A0
Selecting timer mode and functions
b7 b0
0 0 0 0 0 0 0 0
Timer A0 mode register [Address 039616]
TA0MR
Selection of timer mode
Pulse output function select bit
0 : Pulse is not output (TA0OUT pin is a normal port pin)
Gate function select bit
b4 b3
0 0 : Gate function not available (TA0IN pin is a normal port pin)
0 (Must always be "0" in timer mode)
Count source select bit
b7 b6
0 0 : f1
b7 b6 Count source Count source period
f(XIN) : 16MHz f(XCIN) : 32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fC32 976.56µs
Setting counter value
(b15) (b8)
b7 b0 b7 b0
3E16 7F16
Timer A0 register [Address 038716, 038616]
TA0
Setting timer A1
Selecting one-shot timer mode and functions
b7 b0
0 0 0 1 0 1 1 0
Timer A1 mode register [Address 039716]
TA1MR
Selection of one-shot timer mode
Pulse output function select bit
1 : Pulse is output
External trigger select bit (Invalid when choosing timer's overflow as trigger)
Trigger select bit
1 : Selected by event/trigger select register
0 (Must always be "0" in one-shot timer mode)
Count source select bit
b7 b6
0 0 : f1
b7 b6 Count source Count source period
f(XIN) : 16MHz f(XCIN) : 32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fC32 976.56µs
Continued to the next page

Figure 3.2.3. Set-up procedure of variable-period variable-duty PWM output (1)

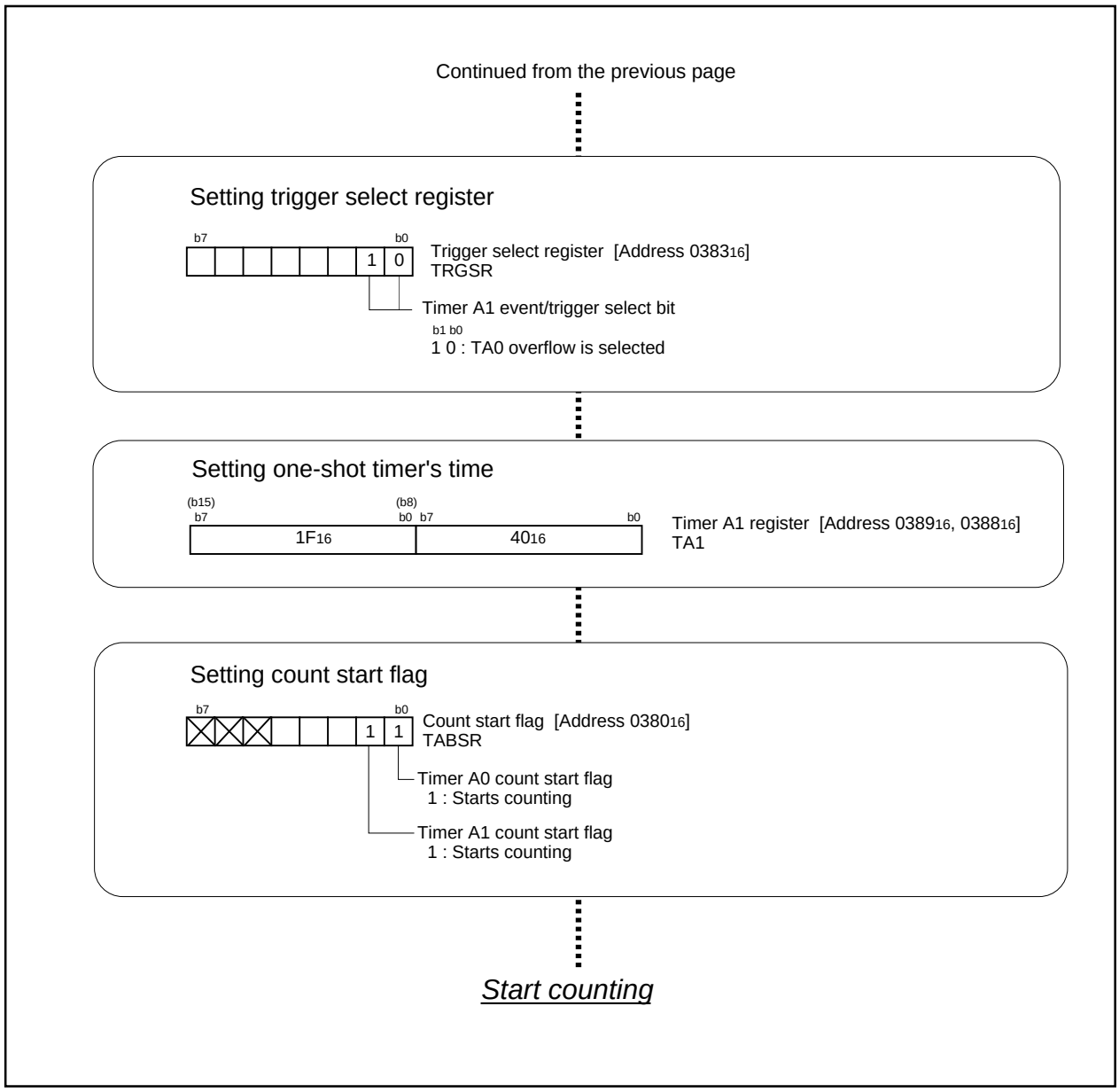


Figure 3.2.4. Set-up procedure of variable-period variable-duty PWM output (2)

3.3 Buzzer Output

Overview The timer mode is used to make the buzzer ring. Figure 3.3.1 shows the operation timing, and Figure 3.3.2 shows the set-up procedure.

Use the following peripheral function:

• The pulse-outputting function in timer mode of timer A.

Specifications

(1) Sound a 2-kHz buzz beep by use of timer A0.

(2) Effect pull-up in the relevant port by use of a pull-up resistor. When the buzzer is off, set the port high-impedance, and stabilize the potential resulting from pulling up.

(3) Connect a 16-MHz oscillator to XIN.

Operation (1) The microcomputer begins performing a count on timer A0. Timer A0 has disabled interrupts.

(2) The microcomputer begins pulse output by setting the pulse output function select bit to "Pulse output effected". P7$_0$ changes into TA0OUT pin and outputs 2-kHz pulses.

(3) The microcomputer stops outputting pulses by setting the pulse output function select bit to "Pulse output not effected". P7$_0$ goes to an input pin, and the output from the pin becomes high-impedance.

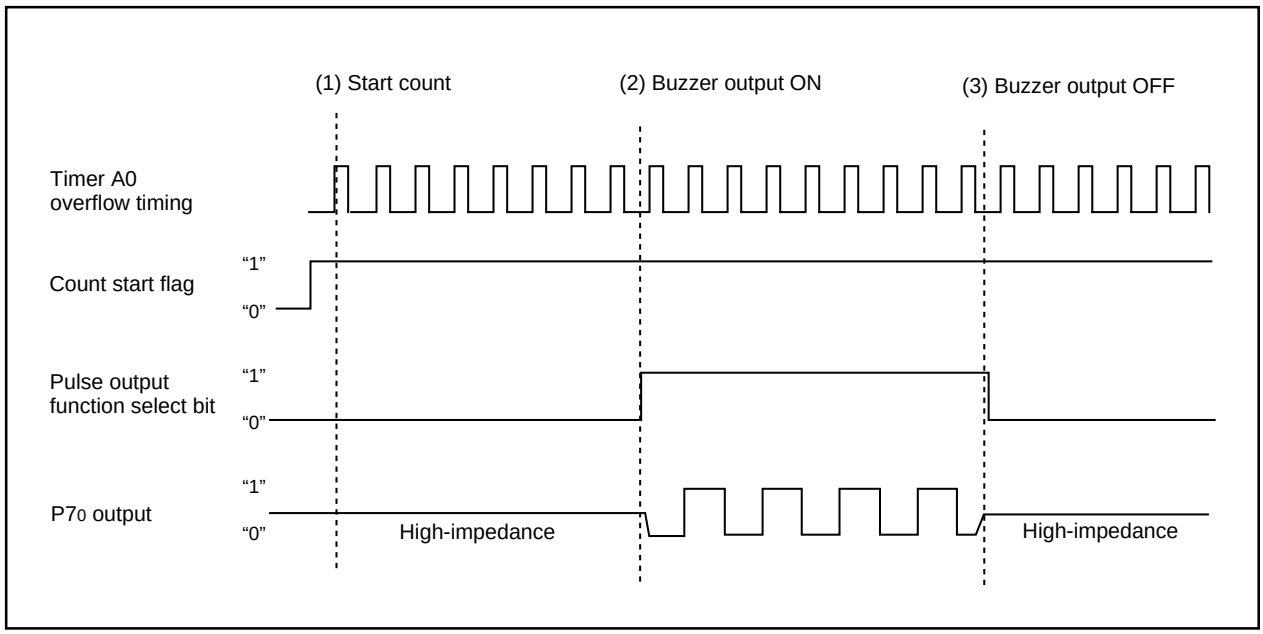


Figure 3.3.1. Operation timing of buzzer output

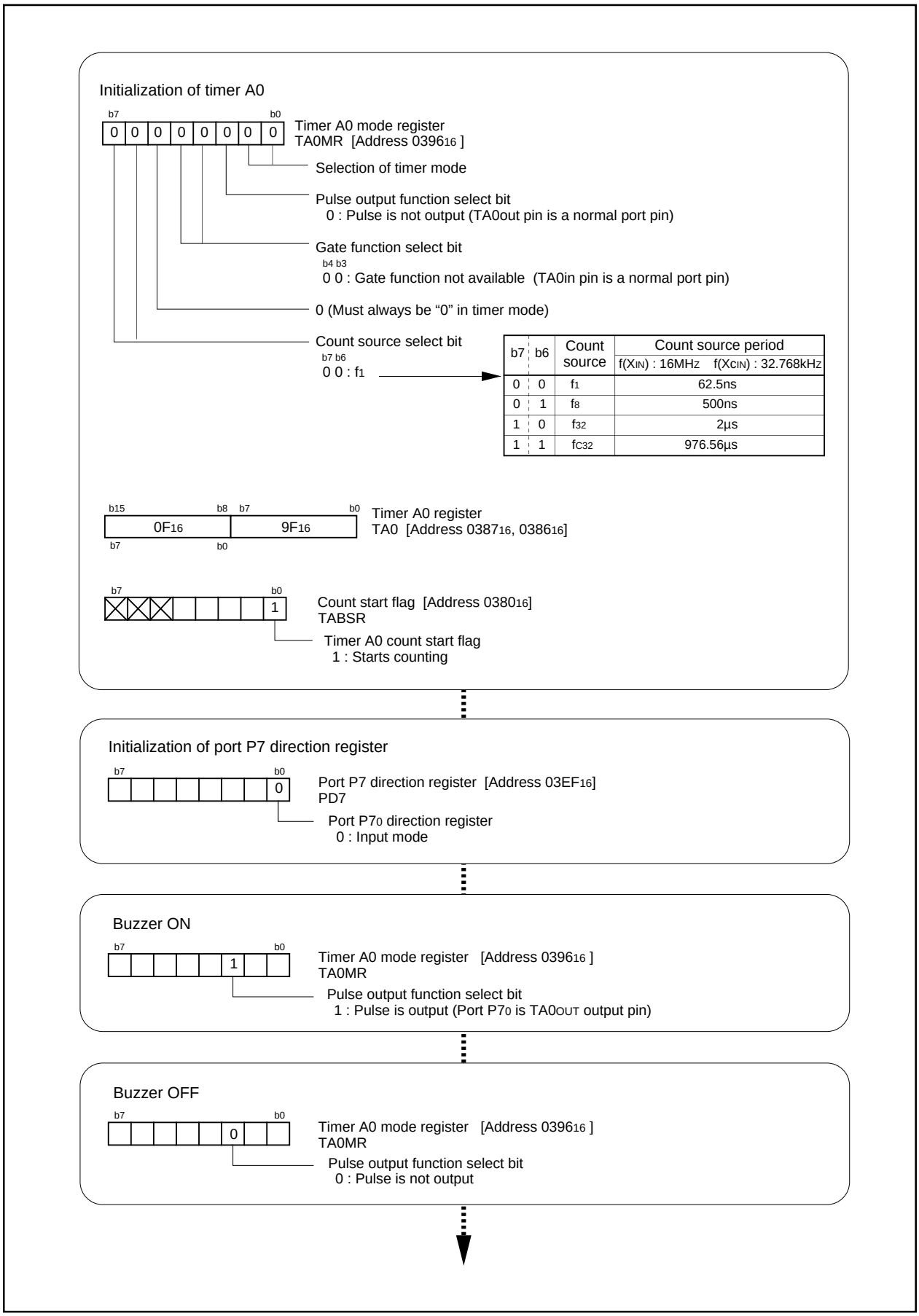

Initialization of timer A0
b7 b0
0 0 0 0 0 0 0 0
Timer A0 mode register
TA0MR [Address 039616]
Selection of timer mode
Pulse output function select bit
0 : Pulse is not output (TA0out pin is a normal port pin)
Gate function select bit
b4 b3
0 0 : Gate function not available (TA0in pin is a normal port pin)
0 (Must always be "0" in timer mode)
Count source select bit
b7 b6
0 0 : f1
b7 b6 Count source Count source period
f(XIN) : 16MHz f(XCIN) : 32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fC32 976.56µs
b15 b8 b7 b0
0F16 9F16
b7 b0
Timer A0 register
TA0 [Address 038716, 038616]
b7 b0
1
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
1 : Starts counting
Initialization of port P7 direction register
b7 b0
0
Port P7 direction register [Address 03EF16]
PD7
Port P70 direction register
0 : Input mode
Buzzer ON
b7 b0
1
Timer A0 mode register [Address 039616]
TA0MR
Pulse output function select bit
1 : Pulse is output (Port P70 is TA0OUT output pin)
Buzzer OFF
b7 b0
0
Timer A0 mode register [Address 039616]
TA0MR
Pulse output function select bit
0 : Pulse is not output

Figure 3.3.2. Set-up procedure of buzzer output

3.4 Solution for External Interrupt Pins Shortage

Overview The following are solution for external interrupt pins shortage. Figure 3.4.1 shows the set-up procedure.
Use the following peripheral function:
• Event counter mode of timer A

Specifications
(1) Inputting a falling edge to the TA0IN pin generates a timer A0 interrupt.

Operation (1) Set timer A0 to event counter mode, set timer to “0”, and set interrupt priority levels in timer A0.
(2) Inputting a falling edge to the TA0IN pin generates a timer A0 interrupt.

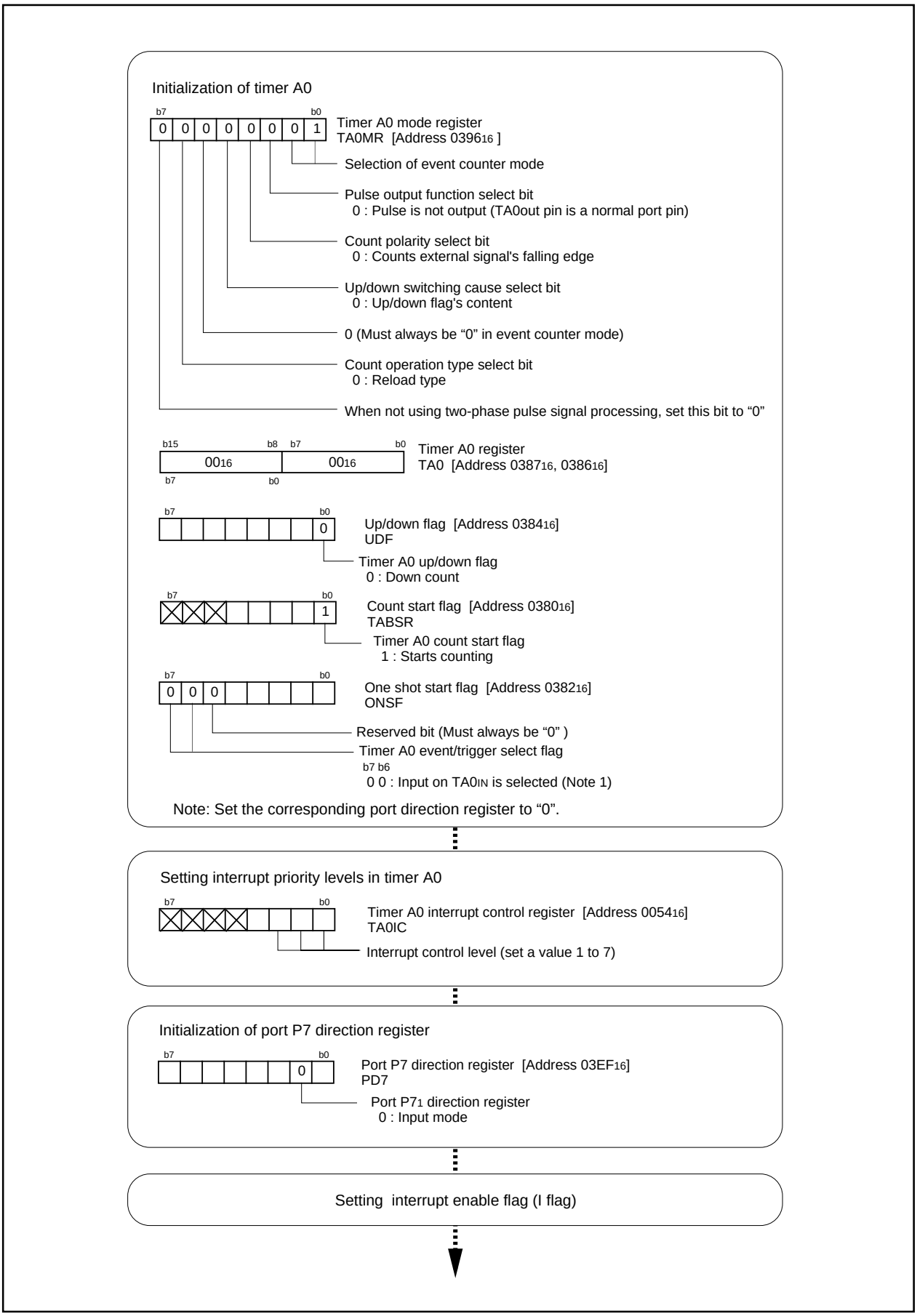

Initialization of timer A0
b7 b0
0 0 0 0 0 0 0 1
Timer A0 mode register
TA0MR [Address 039616]
Selection of event counter mode
Pulse output function select bit
0 : Pulse is not output (TA0out pin is a normal port pin)
Count polarity select bit
0 : Counts external signal's falling edge
Up/down switching cause select bit
0 : Up/down flag's content
0 (Must always be "0" in event counter mode)
Count operation type select bit
0 : Reload type
When not using two-phase pulse signal processing, set this bit to "0"
b15 b8 b7 b0
0016 0016
b7 b0
Timer A0 register
TA0 [Address 038716, 038616]
b7 b0
0
Up/down flag [Address 038416]
UDF
Timer A0 up/down flag
0 : Down count
b7 b0
1
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
1 : Starts counting
b7 b0
0 0 0
One shot start flag [Address 038216]
ONSF
Reserved bit (Must always be "0")
Timer A0 event/trigger select flag
b7 b6
0 0 : Input on TA0IN is selected (Note 1)
Note: Set the corresponding port direction register to "0".
Setting interrupt priority levels in timer A0
b7 b0
Timer A0 interrupt control register [Address 005416]
TA0IC
Interrupt control level (set a value 1 to 7)
Initialization of port P7 direction register
b7 b0
0
Port P7 direction register [Address 03EF16]
PD7
Port P71 direction register
0 : Input mode
Setting interrupt enable flag (I flag)

Figure 3.4.1. Set-up procedure of solution for a shortage of external interrupt pins

3.5 Memory to Memory DMA Transfer

Overview The following are steps for changing both source address and destination address to transfer data from memory to another. The DMA transfer utilizes the workings that assign a higher priority to the DMA0 transfer if transfer requests simultaneously occur in two DMA channels. Figure 3.5.1 shows the operation timing, Figure 3.5.2 shows the block diagram, and Figures 3.5.3 and 3.5.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- Two DMAC channels
- One-byte temporary RAM (address 0800_{16})

Specifications

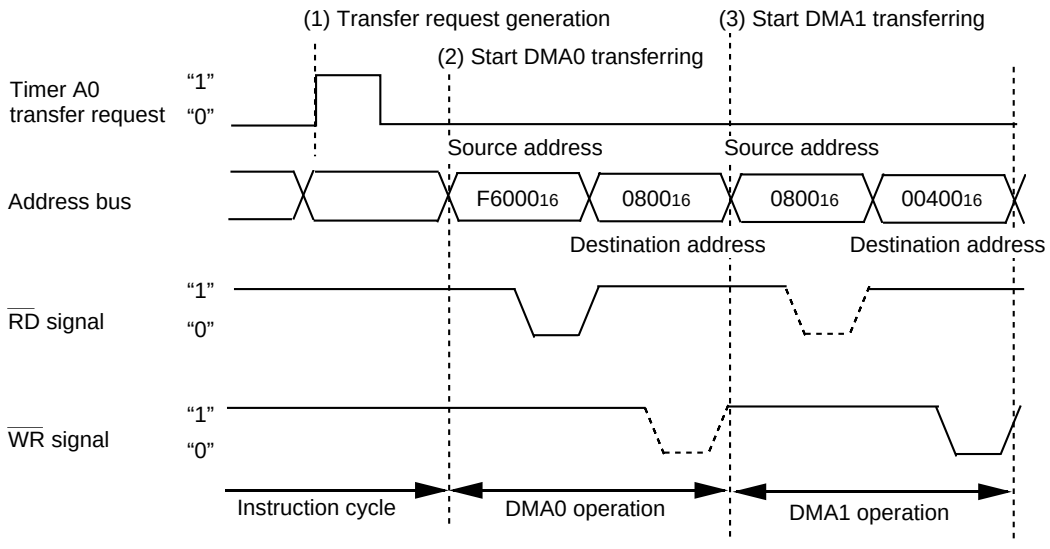
(1) Transfer the content of memory extending over 128 bytes from address $F6000_{16}$ to a 128-byte area starting from address 00400_{16}. Transfer the content every time a timer A0 interrupt request occurs.

(2) Use DMA0 for a transfer from the source to built-in memory, and DMA1 for a transfer from built-in memory to the destination.

Operation (1) A timer A interrupt request occurs. Though both a DMA0 transfer request and a DMA1 transfer request occur simultaneously, the former is executed first.

(2) DMA0 receives a transfer request and transfers data from the source to the built-in memory. At this time, the source address is incremented.

(3) Next, DMA1 receives a transfer request and transfers data involved from built-in memory to the destination. At this time, the destination address is incremented.



Note 1: The DMA0 operation and DMA1 operation are not necessarily executed in succession due to the a cycle steal operation.
Note 2: The instruction cycle varies from instruction to instruction.
Note 3: Since the parts of the $\overline{RD}$ and $\overline{WR}$ signals shown in short-dash lines vary in step with writing to the internal RAM, waveforms are not output to the $\overline{RD}$ and $\overline{WR}$ pins.

Figure 3.5.1. Operation timing of memory to memory DMA transfer

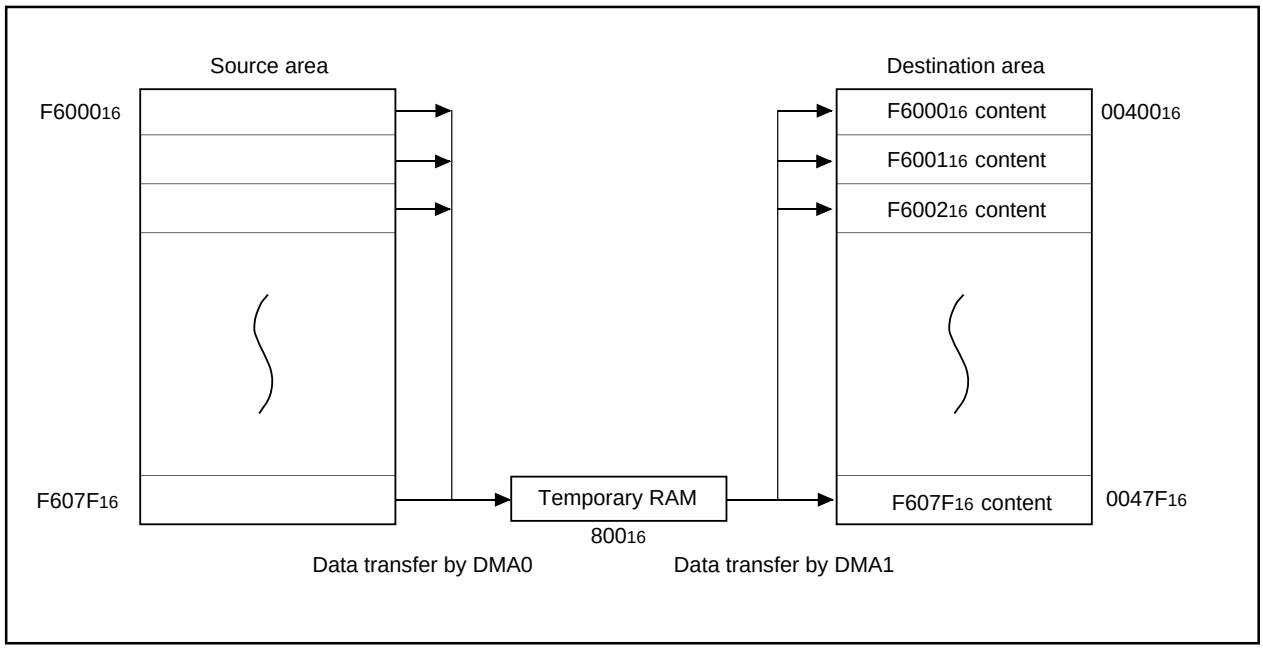


Figure 3.5.2. Block diagram of memory to memory DMA transfer

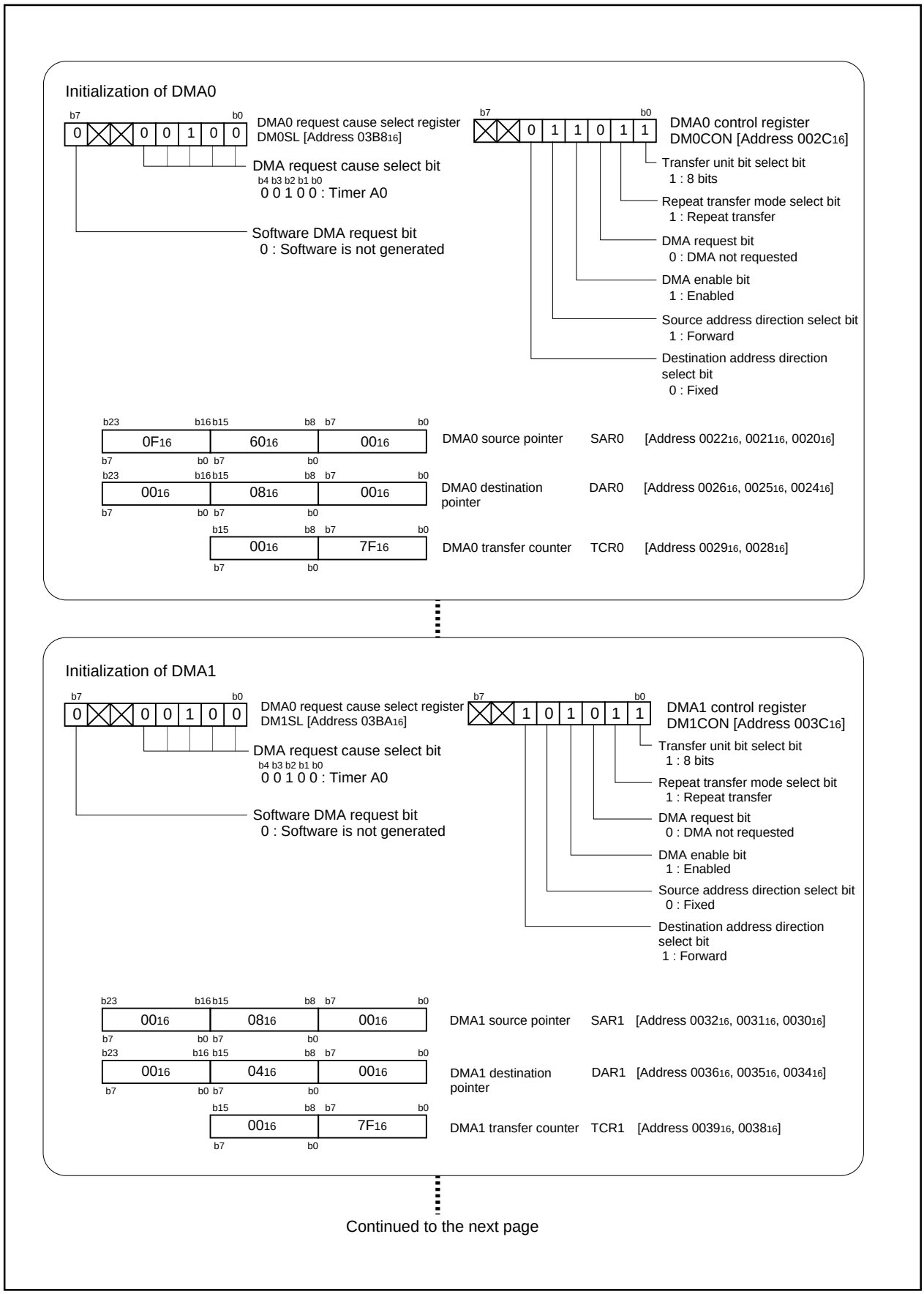

Initialization of DMA0
b7 b0
0 0 0 1 0 0
DMA0 request cause select register
DM0SL [Address 03B816]
DMA request cause select bit
b4 b3 b2 b1 b0
0 0 1 0 0 : Timer A0
Software DMA request bit
0 : Software is not generated
b7 b0
0 1 1 0 1 1
DMA0 control register
DM0CON [Address 002C16]
Transfer unit bit select bit
1 : 8 bits
Repeat transfer mode select bit
1 : Repeat transfer
DMA request bit
0 : DMA not requested
DMA enable bit
1 : Enabled
Source address direction select bit
1 : Forward
Destination address direction
select bit
0 : Fixed
b23 b16 b15 b8 b7 b0
0F16 6016 0016
DMA0 source pointer SAR0 [Address 002216, 002116, 002016]
0016 0816 0016
DMA0 destination
pointer
DAR0 [Address 002616, 002516, 002416]
0016 7F16
DMA0 transfer counter TCR0 [Address 002916, 002816]
Initialization of DMA1
b7 b0
0 0 0 1 0 0
DMA0 request cause select register
DM1SL [Address 03BA16]
DMA request cause select bit
b4 b3 b2 b1 b0
0 0 1 0 0 : Timer A0
Software DMA request bit
0 : Software is not generated
b7 b0
1 0 1 0 1 1
DMA1 control register
DM1CON [Address 003C16]
Transfer unit bit select bit
1 : 8 bits
Repeat transfer mode select bit
1 : Repeat transfer
DMA request bit
0 : DMA not requested
DMA enable bit
1 : Enabled
Source address direction select bit
0 : Fixed
Destination address direction
select bit
1 : Forward
0016 0816 0016
DMA1 source pointer SAR1 [Address 003216, 003116, 003016]
0016 0416 0016
DMA1 destination
pointer
DAR1 [Address 003616, 003516, 003416]
0016 7F16
DMA1 transfer counter TCR1 [Address 003916, 003816]
Continued to the next page

Figure 3.5.3. Set-up procedure of memory to memory DMA transfer (1)

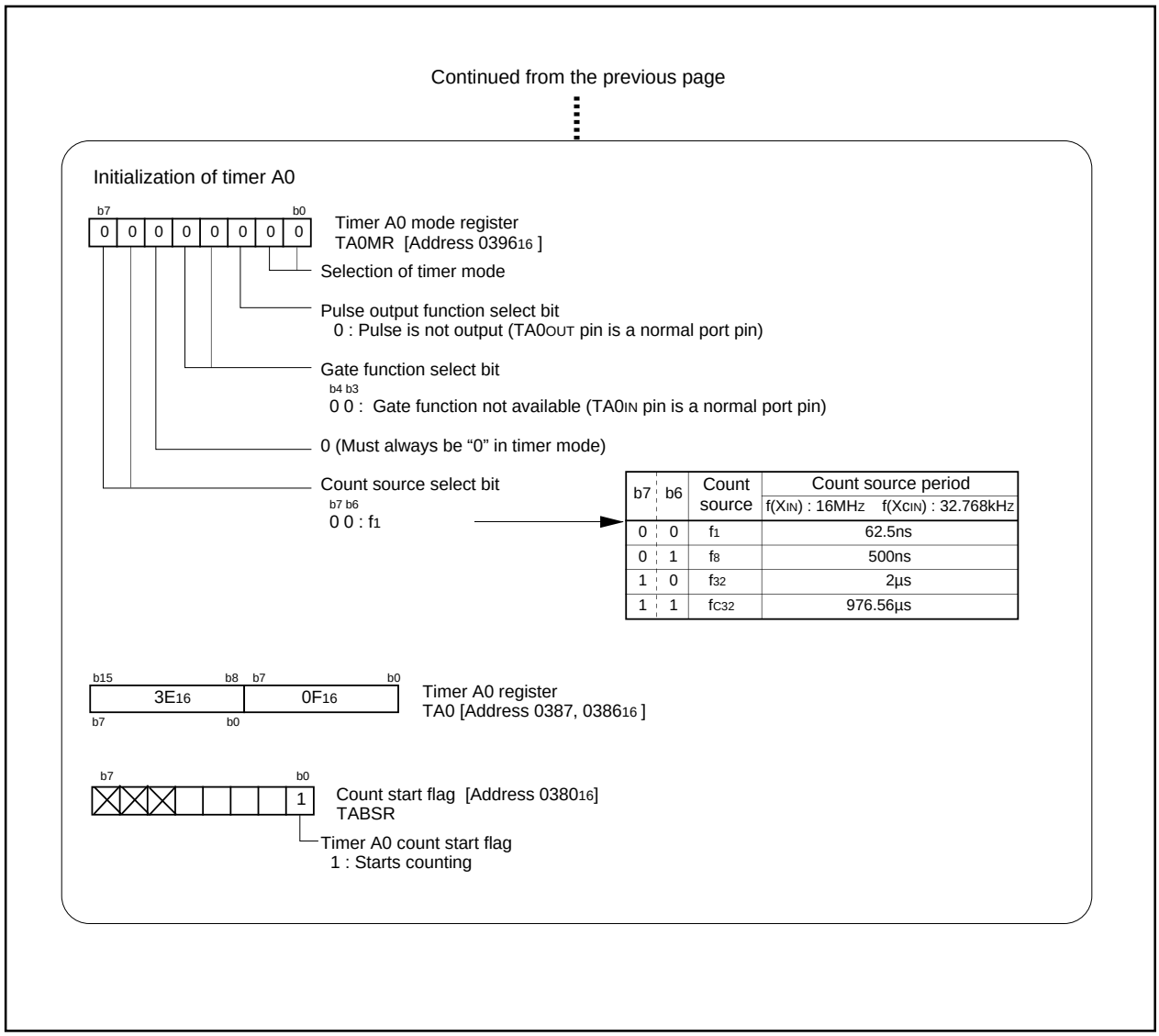

Continued from the previous page
Initialization of timer A0
b7 b0
0 0 0 0 0 0 0 0
Timer A0 mode register
TA0MR [Address 039616]
Selection of timer mode
Pulse output function select bit
0 : Pulse is not output (TA0OUT pin is a normal port pin)
Gate function select bit
b4 b3
0 0 : Gate function not available (TA0IN pin is a normal port pin)
0 (Must always be "0" in timer mode)
Count source select bit
b7 b6
0 0 : f1
b7 b6 Count source Count source period
f(XIN) : 16MHz f(XCIN) : 32.768kHz
0 0 f1 62.5ns
0 1 f8 500ns
1 0 f32 2µs
1 1 fC32 976.56µs
b15 b8 b7 b0
3E16 0F16
b7 b0
Timer A0 register
TA0 [Address 0387, 038616]
b7 b0
1
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
1 : Starts counting

Figure 3.5.4. Set-up procedure of memory to memory DMA transfer (2)

3.6 CRC Calculation SFR Access Snoop Function in Clock Synchronous Serial Data Transmit

Overview The M30245 group, by use of DMAC, transfers data from the internal RAM to the UART1 and the result is transferred to the UART1 by use of SFR access snoop function. The block diagram is shown in Figure 3.6.1 and the setting routine is shown in Figure 3.6.2 to Figure 3.6.4.

The peripheral functions to be used are as follows:

- DMAC 1 Channel
- Internal RAM (address 0040_{16}) 512 bytes
- UART1 (Clock synchronous serial I/O mode)
- CRC calculation circuit
- SFR access snoop function

Specifications

(1) Data transfer is performed starting at address 0040_{16} from the area with 512 bytes to the UART1. Data are transferred from area between the address 0040_{16} and the 512nd byte to the UART1. Transfer is executed every time 1 byte of serial transmit is completed.

(2) Use the DMA0 to transfer data from the internal RAM to the UART1. Select the UART1 transmit to the DMA0 request factor. Select the single transfer mode and set the DMA0 transfer counter to 511 bytes (512-1).

(3) Set the CRC calculation circuit to the CRC-CCITT and set CRC snoop address register to the address of UART1 transmit buffer register (write snoop).

(4) On completing the DMA, 2-byte data of CRC data register (calculation result) are transferred to the UART1 and operation is completed.

Operation

(1) Initialize the UART1 related registers.

(2) Initialize the DMA0 related registers in DMA disable state.

(3) Set the DMA0 transfer counter to the transfer data consisting of 511 bytes (in this case, 8-bit transfer).

(4) Initialize the CRC calculation circuit and the SFR access snoop function.

(5) Set the software DMA request bit of DMA0 to “1”. At this time, 1st byte data are transferred from RAM to the transmit buffer of the UART1. Simultaneously, the transfer source address is incremented and the content of the transfer counter is down-counted. The transferred data are automatically written in CRC input register by the SFR access snoop function.

(6) When the transmit buffer of the UART1 becomes writable state, the DMA transfer request is occurred by the UART1. At this time, the next data are transferred from RAM to the transmit buffer of the UART1. Simultaneously, the transfer source address is incremented and the content of the transfer counter is down-counted. The transferred data are automatically written in CRC input register by the SFR access snoop function.

(7) As a result of repetition of the above (6), when the DMA0 transfer counter underflow, DMA enable bit is set to “0” to complete the DMA0 transfer. Simultaneously, the DMA0 interrupt request occurs. When the DMA0 interrupt request is detected, CRC data register (2 bytes) is read, it is transferred to the UART1 transmit buffer sequentially.

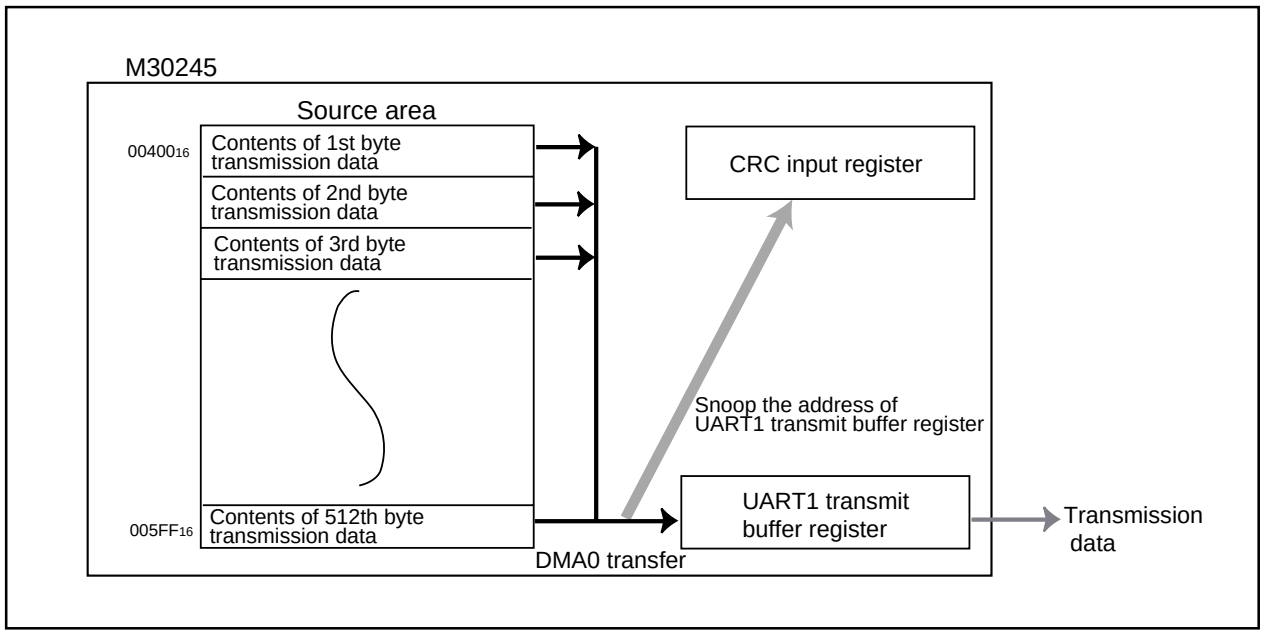


Figure 3.6.1. Block diagram of DMA transfer from RAM to UART and SFR snooping function

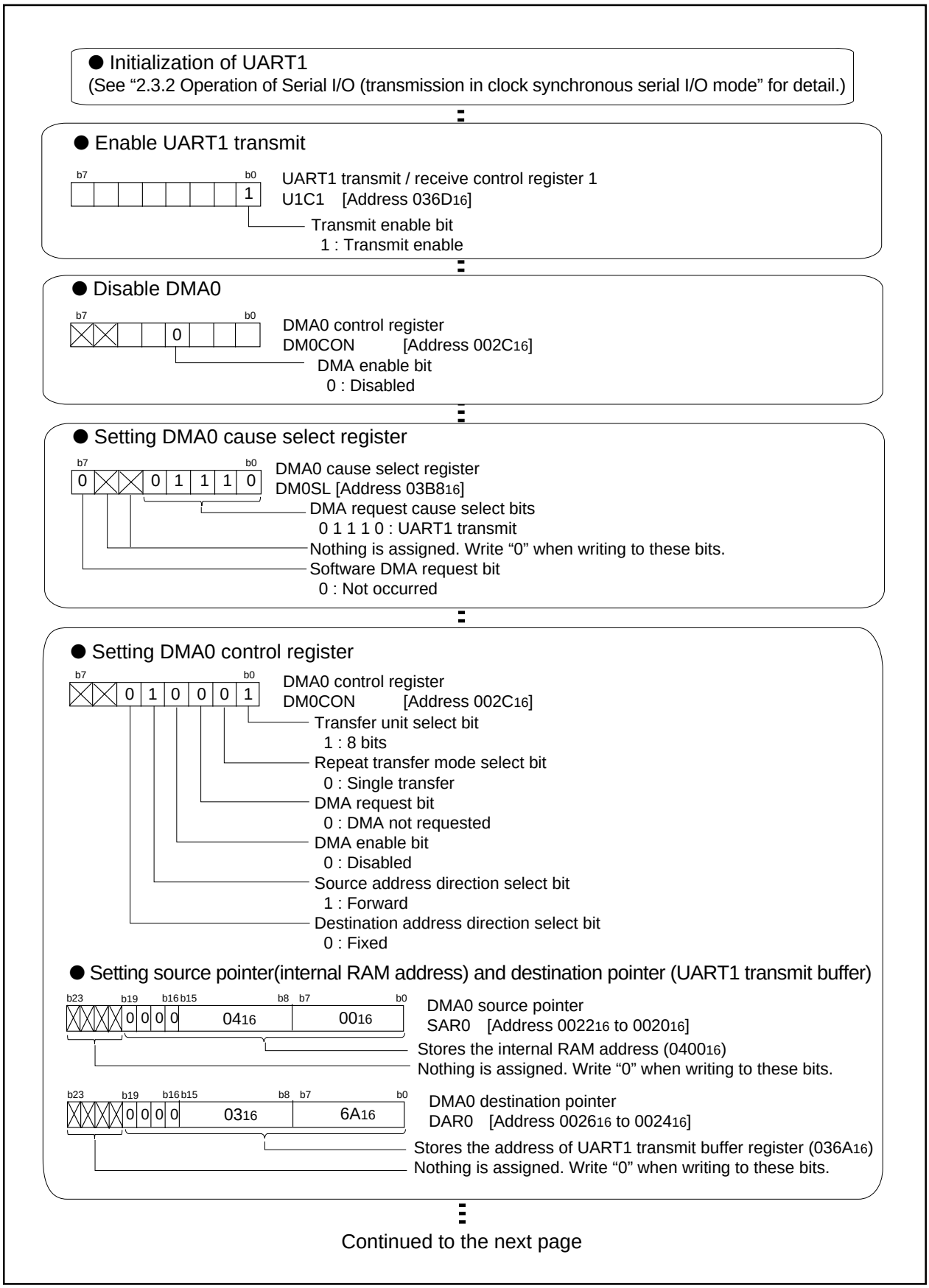

● Initialization of UART1
(See “2.3.2 Operation of Serial I/O (transmission in clock synchronous serial I/O mode” for detail.)
● Enable UART1 transmit
b7 b0
1
UART1 transmit / receive control register 1
U1C1 [Address 036D16]
Transmit enable bit
1 : Transmit enable
● Disable DMA0
b7 b0
0
DMA0 control register
DM0CON [Address 002C16]
DMA enable bit
0 : Disabled
● Setting DMA0 cause select register
b7 b0
0 0 1 1 1 0
DMA0 cause select register
DM0SL [Address 03B816]
DMA request cause select bits
0 1 1 1 0 : UART1 transmit
Nothing is assigned. Write “0” when writing to these bits.
Software DMA request bit
0 : Not occurred
● Setting DMA0 control register
b7 b0
0 1 0 0 0 1
DMA0 control register
DM0CON [Address 002C16]
Transfer unit select bit
1 : 8 bits
Repeat transfer mode select bit
0 : Single transfer
DMA request bit
0 : DMA not requested
DMA enable bit
0 : Disabled
Source address direction select bit
1 : Forward
Destination address direction select bit
0 : Fixed
● Setting source pointer(internal RAM address) and destination pointer (UART1 transmit buffer)
b23 b19 b16 b15 b8 b7 b0
0 0 0 0 0416 0016
DMA0 source pointer
SAR0 [Address 002216 to 002016]
Stores the internal RAM address (040016)
Nothing is assigned. Write “0” when writing to these bits.
b23 b19 b16 b15 b8 b7 b0
0 0 0 0 0316 6A16
DMA0 destination pointer
DAR0 [Address 002616 to 002416]
Stores the address of UART1 transmit buffer register (036A16)
Nothing is assigned. Write “0” when writing to these bits.
Continued to the next page

Figure 3.6.2. Setting routine (1) of DMA transfer from RAM to UART using SFR snooping function

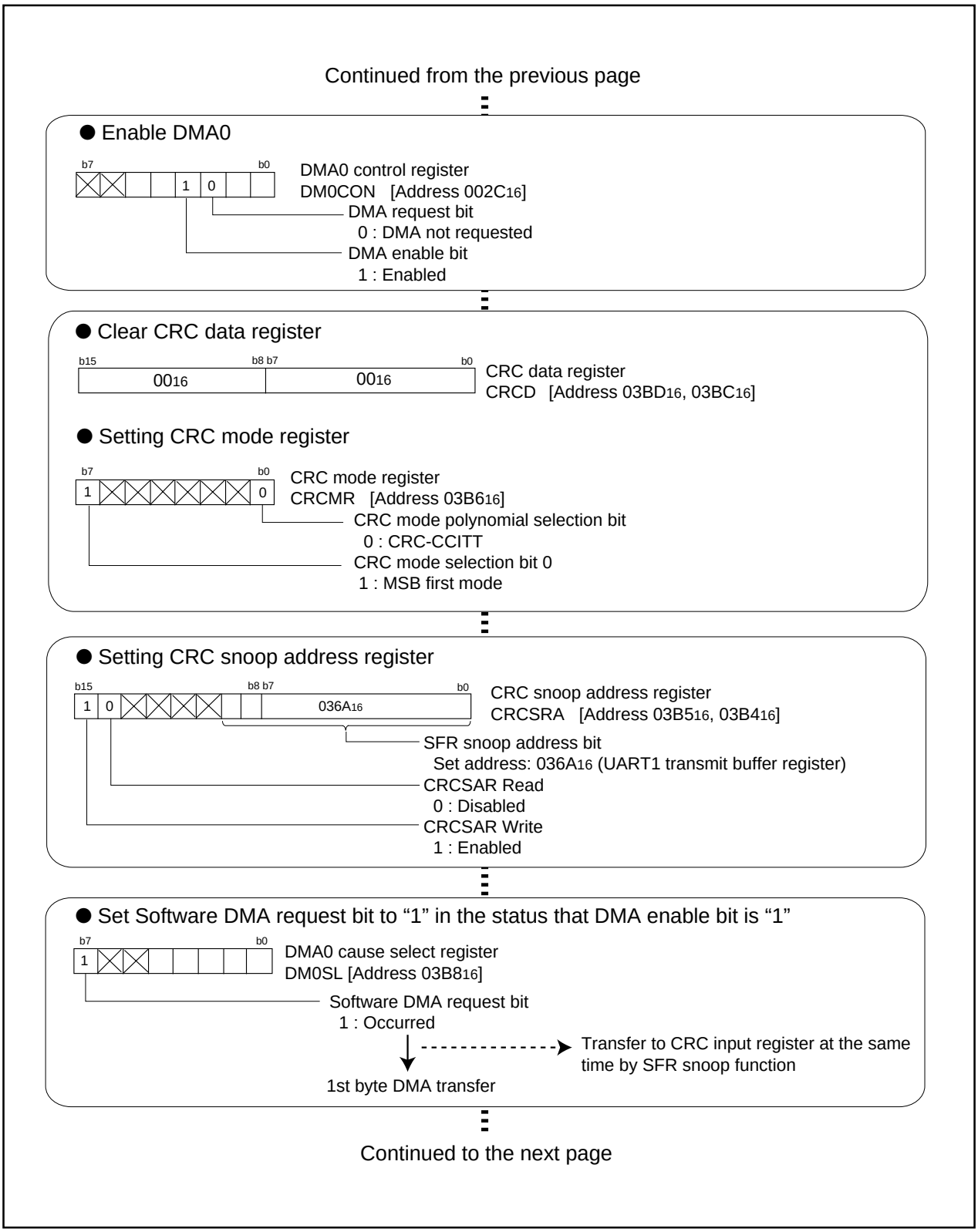

Continued from the previous page
● Enable DMA0
b7 b0
1 0
DMA0 control register
DM0CON [Address 002C16]
DMA request bit
0 : DMA not requested
DMA enable bit
1 : Enabled
● Clear CRC data register
b15 b8 b7 b0
0016 0016
CRC data register
CRCD [Address 03BD16, 03BC16]
● Setting CRC mode register
b7 b0
1 0
CRC mode register
CRCMR [Address 03B616]
CRC mode polynomial selection bit
0 : CRC-CCITT
CRC mode selection bit 0
1 : MSB first mode
● Setting CRC snoop address register
b15 b8 b7 b0
1 0 036A16
CRC snoop address register
CRCSRA [Address 03B516, 03B416]
SFR snoop address bit
Set address: 036A16 (UART1 transmit buffer register)
CRCSAR Read
0 : Disabled
CRCSAR Write
1 : Enabled
● Set Software DMA request bit to “1” in the status that DMA enable bit is “1”
b7 b0
1
DMA0 cause select register
DM0SL [Address 03B816]
Software DMA request bit
1 : Occurred
Transfer to CRC input register at the same time by SFR snoop function
1st byte DMA transfer
Continued to the next page

Figure 3.6.3. Setting routine (2) of DMA transfer from RAM to UART using SFR snooping function

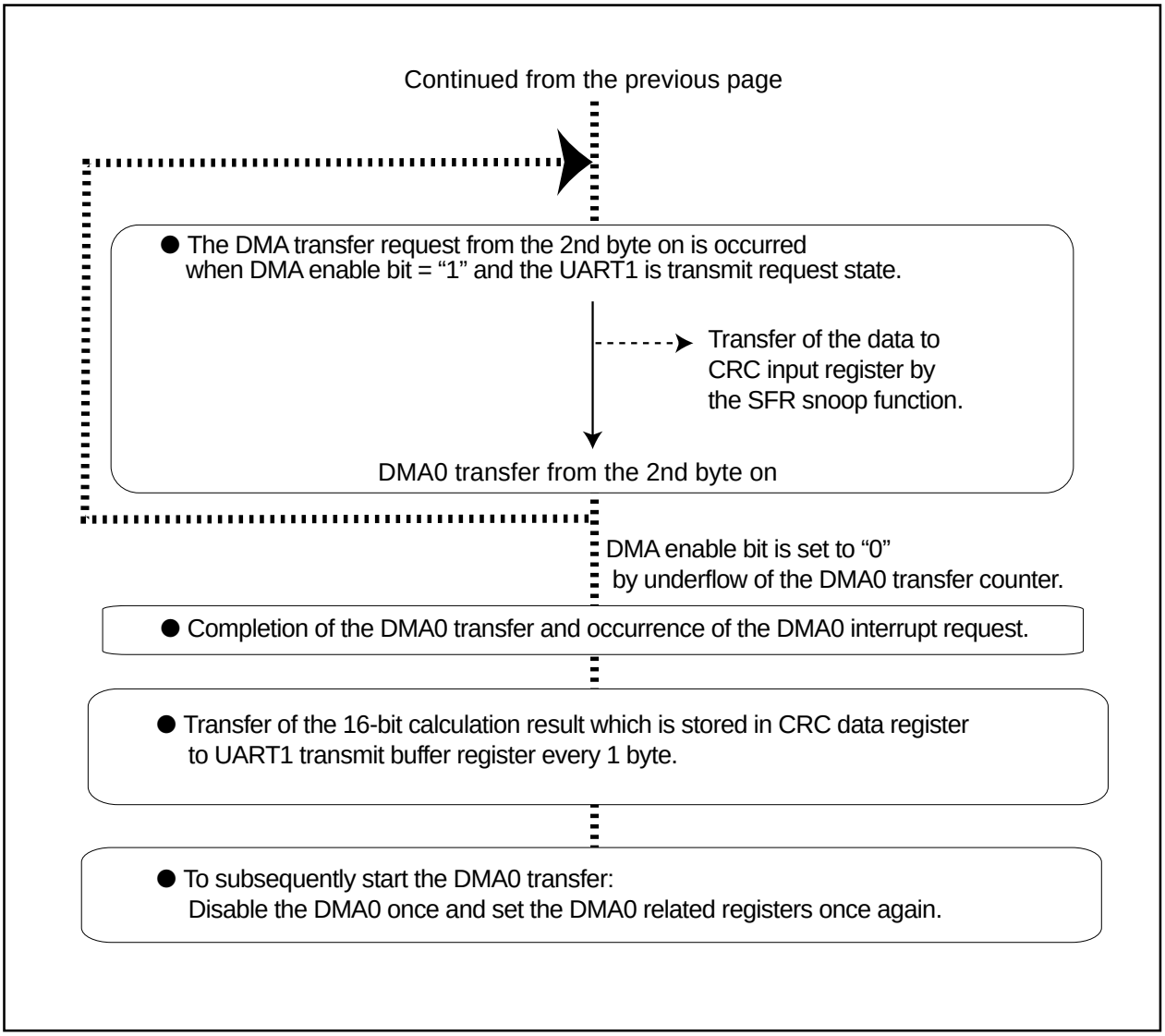

Continued from the previous page
● The DMA transfer request from the 2nd byte on is occurred when DMA enable bit = “1” and the UART1 is transmit request state.
Transfer of the data to CRC input register by the SFR snoop function.
DMA0 transfer from the 2nd byte on
DMA enable bit is set to “0” by underflow of the DMA0 transfer counter.
● Completion of the DMA0 transfer and occurrence of the DMA0 interrupt request.
● Transfer of the 16-bit calculation result which is stored in CRC data register to UART1 transmit buffer register every 1 byte.
● To subsequently start the DMA0 transfer:
Disable the DMA0 once and set the DMA0 related registers once again.

Figure 3.6.4. Setting routine (3) of DMA transfer from RAM to UART using SFR snooping function

3.7 Transfer from USB FIFO to Serial Sound Interface

Overview The M30245 group, by use of DMAC, transfers data from the USB endpoint 1 OUT FIFO to SS interface 1 transmit buffer register and fetches one packet data.

The block diagram is shown in Figure 3.7.1 and the setting routine is shown in Figure 3.7.2 to Figure 3.7.4.

The peripheral functions to be used are as follows:

- DMAC 1 channel
- USB endpoint 1 OUT (Receive)
- Serial sound interface 1

Specifications

(1) Receive packet data of the endpoint 1 OUT FIFO are transferred to SS interface 1 transmit buffer register. Transfer is executed every time the DMA transfer factor of the serial sound interface 1 occurs.

(2) Use the DMA0 to transfer data from the endpoint 1 OUT FIFO to SS interface 1 transmit buffer register. Select the serial sound interface 1 transmit to the DMA0 request factor. Select the single transfer mode and set the DMA0 transfer counter to 1/2 × (the data count of one packet received with endpoint 1 OUT) –1.

(3) Set the endpoint 1 OUT maximum packet size to 288 bytes (when sampling 48KHz/ 24-bit/ stereo) and disable the AUTO_CLR function. The data count of receive packet of endpoint 1 (endpoint 1 OUT write count register) is set to 288 bytes. Endpoint 1 OUT is used in isochronous transfer.

(4) On completing the DMA0 transfer, fetch of one packet data from the endpoint 1 OUT FIFO is completed by setting CLR_OUT_BUF_RDY bit of endpoint 1 to “1”.

Operation

(1) Initialize the DMA0 related registers in the state which DMA is disabled and USB DMA0 request register is not selected (in this case, 16-bit transfer).

(2) When the OUT_BUF_STS1 flag of endpoint 1 is set to “1” and packet data receive has been detected, set the DMA0 transfer counter to the 1/2 × (the data count of receive one packet) –1 (in this application example, 143 value is set).

(3) Set DMA enable bit of DMA0CON to “1” (DMA0 is enabled). Then, the DMA0 transfer request from the serial sound interface occurs.

(4) When the transfer request is received, the DMA0 transfers the 1st word (16-bit) data from the endpoint 1 OUT FIFO to the serial sound interface 1. Simultaneously, the content of the transfer counter is down-counted. Then, the DMA0 transfer request from the serial sound interface occurs .

(5) As a result of repetition of the above (4), when the DMA0 transfer counter underflow, DMA enable bit is set to “0” to complete the DMA0 transfer. Simultaneously, the DMA0 interrupt request occurs. When the DMA0 interrupt request is detected, set CLR_OUT_BUF_RDY bit of endpoint 1 OUT to “1”.

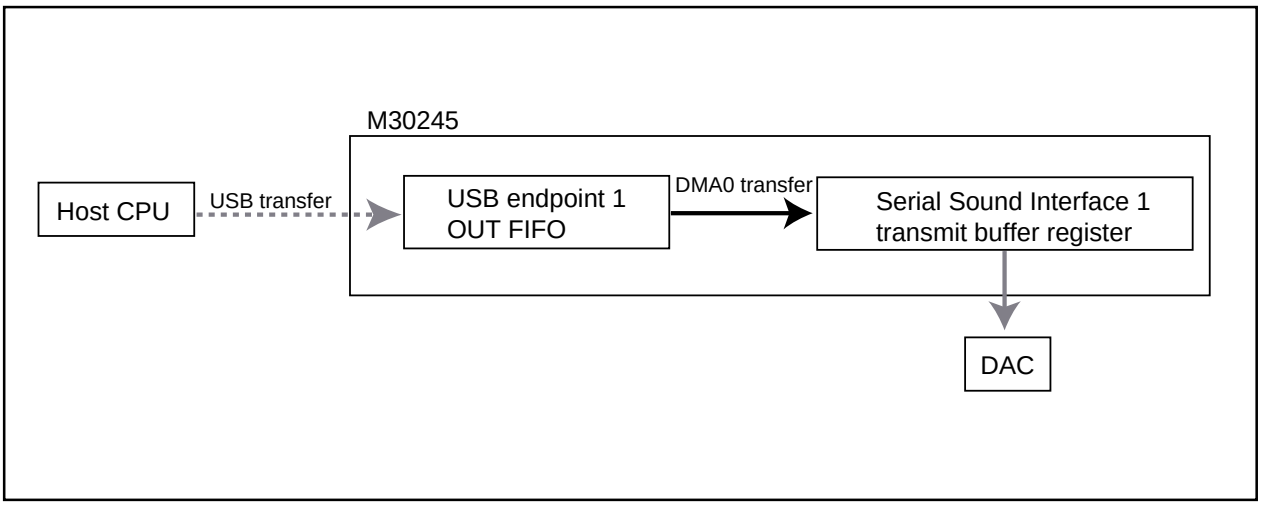


Figure 3.7.1. Block diagram of DMA transfer from USB FIFO to serial sound interface

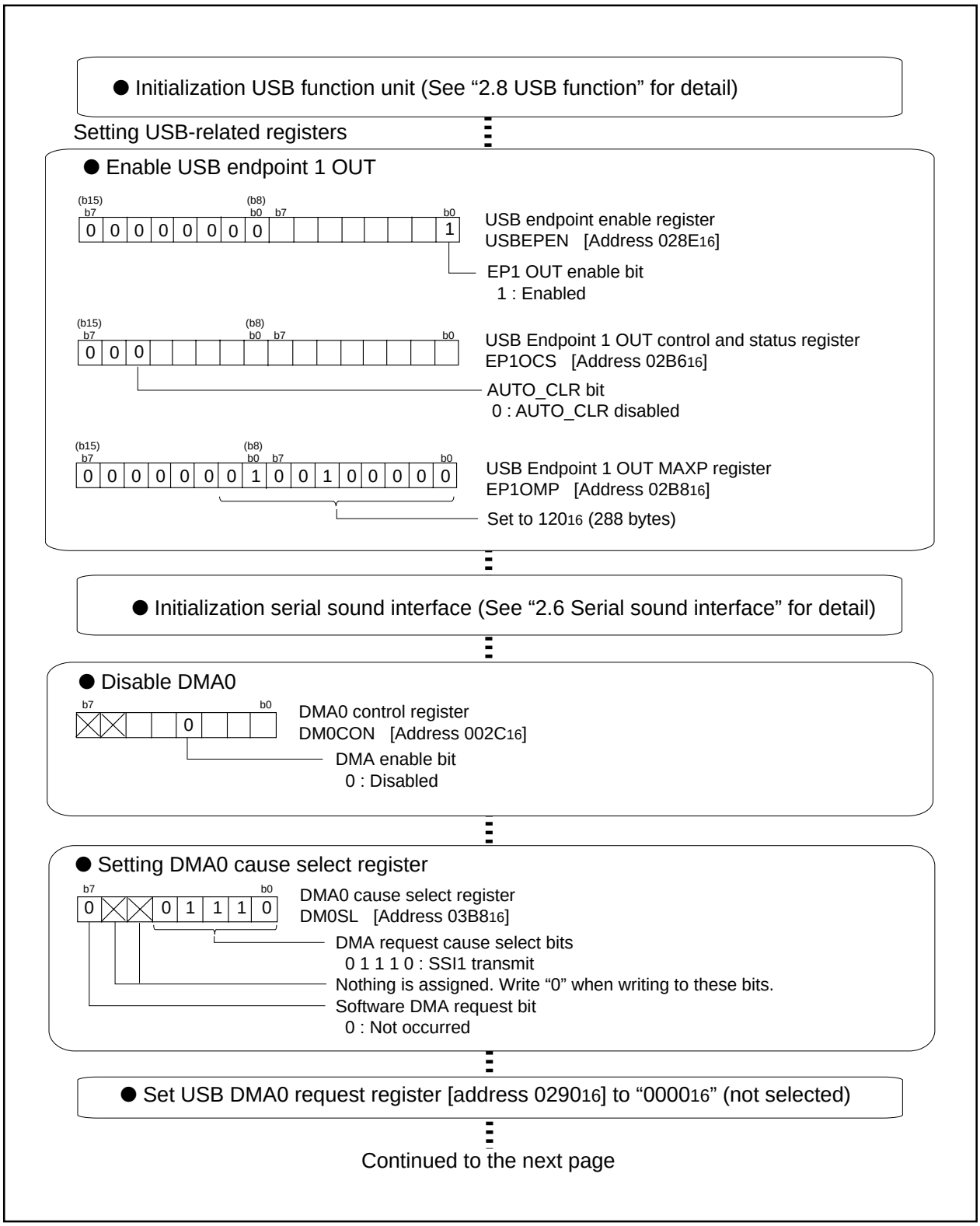

● Initialization USB function unit (See “2.8 USB function” for detail)
Setting USB-related registers
● Enable USB endpoint 1 OUT
(b15) (b8)
b7 b0 b7 b0
0 0 0 0 0 0 0 0 1
USB endpoint enable register
USBEPEN [Address 028E16]
EP1 OUT enable bit
1 : Enabled
(b15) (b8)
b7 b0 b7 b0
0 0 0
USB Endpoint 1 OUT control and status register
EP1OCS [Address 02B616]
AUTO_CLR bit
0 : AUTO_CLR disabled
(b15) (b8)
b7 b0 b7 b0
0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0
USB Endpoint 1 OUT MAXP register
EP1OMP [Address 02B816]
Set to 12016 (288 bytes)
● Initialization serial sound interface (See “2.6 Serial sound interface” for detail)
● Disable DMA0
b7 b0
0
DMA0 control register
DM0CON [Address 002C16]
DMA enable bit
0 : Disabled
● Setting DMA0 cause select register
b7 b0
0 0 1 1 1 0
DMA0 cause select register
DM0SL [Address 03B816]
DMA request cause select bits
0 1 1 1 0 : SSI1 transmit
Nothing is assigned. Write “0” when writing to these bits.
Software DMA request bit
0 : Not occurred
● Set USB DMA0 request register [address 029016] to “000016” (not selected)
Continued to the next page

Figure 3.7.2. Setting routine (1) of DMA transfer from USB OUT FIFO to serial sound interface

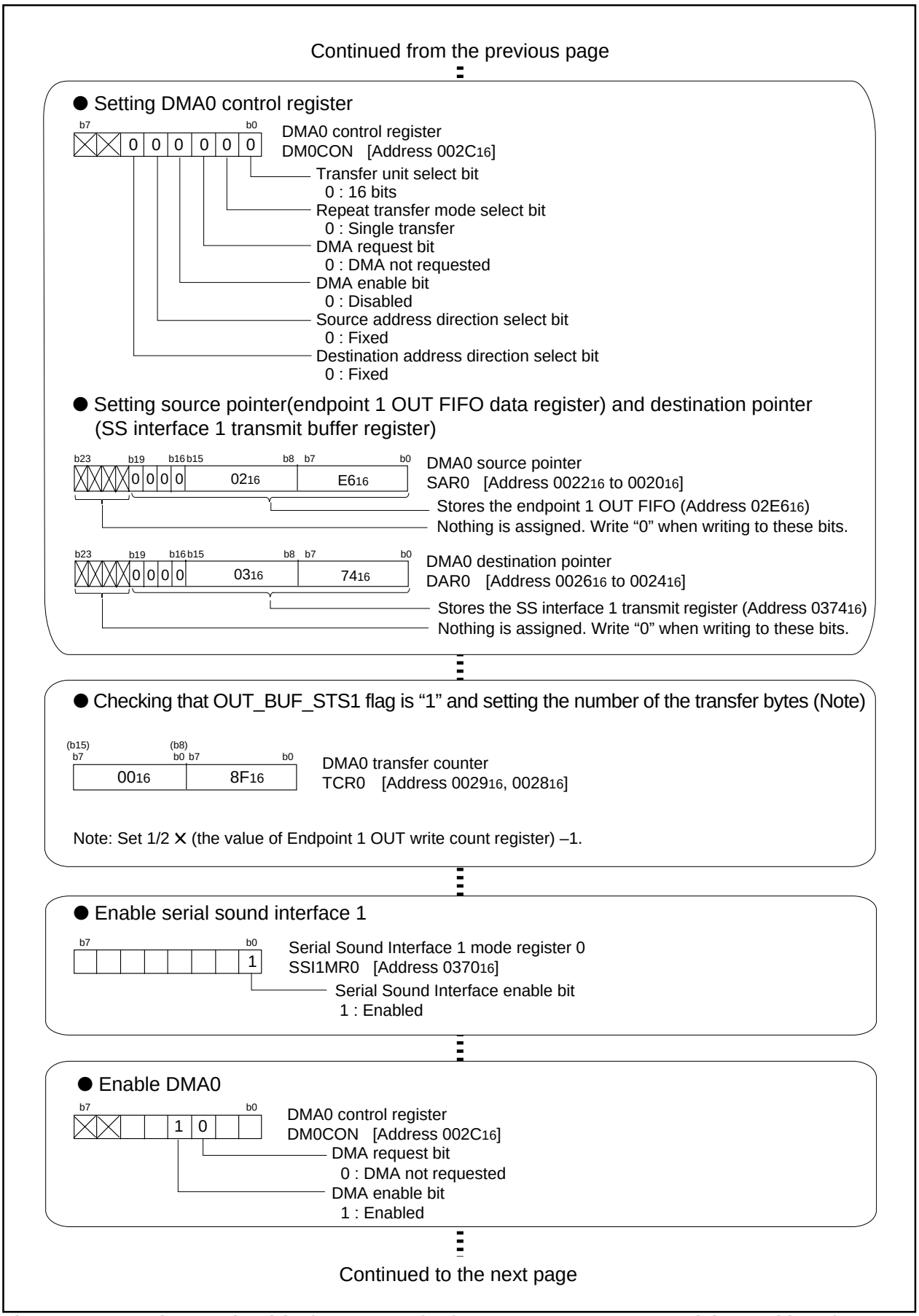

Continued from the previous page
● Setting DMA0 control register
b7 b0
0 0 0 0 0 0
DMA0 control register
DM0CON [Address 002C16]
Transfer unit select bit
0 : 16 bits
Repeat transfer mode select bit
0 : Single transfer
DMA request bit
0 : DMA not requested
DMA enable bit
0 : Disabled
Source address direction select bit
0 : Fixed
Destination address direction select bit
0 : Fixed
● Setting source pointer(endpoint 1 OUT FIFO data register) and destination pointer (SS interface 1 transmit buffer register)
b23 b19 b16 b15 b8 b7 b0
0 0 0 0 0216 E616
DMA0 source pointer
SAR0 [Address 002216 to 002016]
Stores the endpoint 1 OUT FIFO (Address 02E616)
Nothing is assigned. Write “0” when writing to these bits.
b23 b19 b16 b15 b8 b7 b0
0 0 0 0 0316 7416
DMA0 destination pointer
DAR0 [Address 002616 to 002416]
Stores the SS interface 1 transmit register (Address 037416)
Nothing is assigned. Write “0” when writing to these bits.
● Checking that OUT_BUF_STS1 flag is “1” and setting the number of the transfer bytes (Note)
(b15) (b8)
b7 b0 b7 b0
0016 8F16
DMA0 transfer counter
TCR0 [Address 002916, 002816]
Note: Set 1/2 × (the value of Endpoint 1 OUT write count register) –1.
● Enable serial sound interface 1
b7 b0
1
Serial Sound Interface 1 mode register 0
SSI1MR0 [Address 037016]
Serial Sound Interface enable bit
1 : Enabled
● Enable DMA0
b7 b0
1 0
DMA0 control register
DM0CON [Address 002C16]
DMA request bit
0 : DMA not requested
DMA enable bit
1 : Enabled
Continued to the next page

Figure 3.7.3. Setting routine (2) of DMA transfer from USB OUT FIFO to serial sound interface

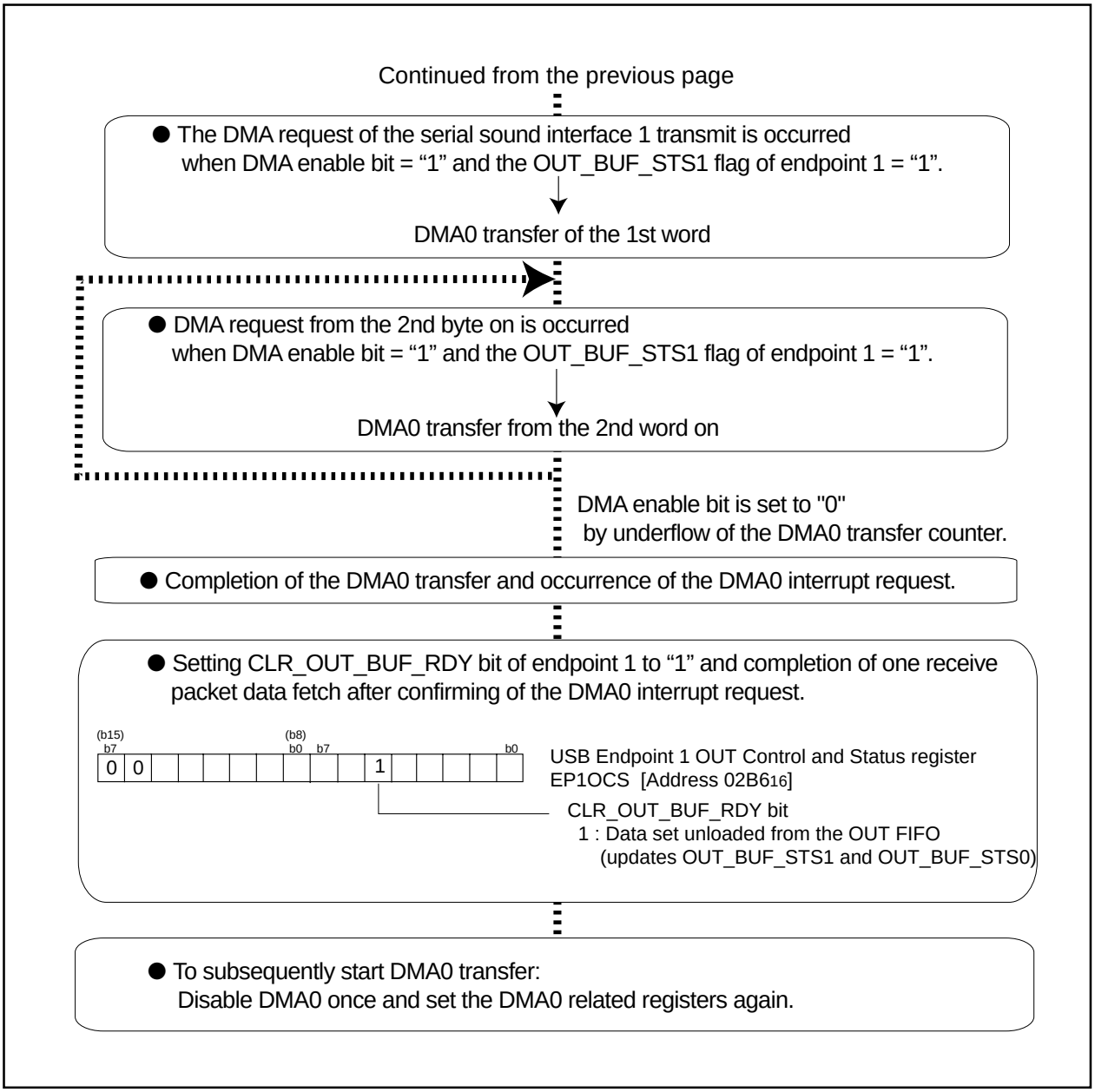

Continued from the previous page
● The DMA request of the serial sound interface 1 transmit is occurred when DMA enable bit = “1” and the OUT_BUF_STS1 flag of endpoint 1 = “1”.
DMA0 transfer of the 1st word
● DMA request from the 2nd byte on is occurred when DMA enable bit = “1” and the OUT_BUF_STS1 flag of endpoint 1 = “1”.
DMA0 transfer from the 2nd word on
DMA enable bit is set to "0" by underflow of the DMA0 transfer counter.
● Completion of the DMA0 transfer and occurrence of the DMA0 interrupt request.
● Setting CLR_OUT_BUF_RDY bit of endpoint 1 to “1” and completion of one receive packet data fetch after confirming of the DMA0 interrupt request.
(b15)
b7
(b8)
b0
b7
b0
0 0 1
USB Endpoint 1 OUT Control and Status register
EP1OCS [Address 02B616]
CLR_OUT_BUF_RDY bit
1 : Data set unloaded from the OUT FIFO
(updates OUT_BUF_STS1 and OUT_BUF_STS0)
● To subsequently start DMA0 transfer:
Disable DMA0 once and set the DMA0 related registers again.

Figure 3.7.4. Setting routine (3) of DMA transfer from USB OUT FIFO to serial sound interface

3.8 Controlling Power Using Stop Mode

Overview The following are steps for controlling power using stop mode. Figure 3.8.1 shows the operation timing, Figure 3.8.2 shows an example of circuit, and Figures 3.8.3 and 3.8.4 show the set-up procedure.

Use the following peripheral functions:

- Key-input interrupts
- Stop mode
- Pull-up function

This example is not performed USB power control. Please refer section 2.8.4 for the power control of USB related.

Specifications

(1) Use $P0_0$ through $P0_3$ for the scan output pins of a key matrix. Use the input pins ($\overline{KI_0}$ through $\overline{KI_3}$) of the key-input interrupt function for the key-input reading pins. The pull-up function is also used.

(2) If a key-input interrupt request occurs, clear the stop mode and read a key.

Operation (1) Enable a key-input interrupt and set the pull-up function to pins $\overline{KI_0}$ through $\overline{KI_3}$. Change the output of $P0_0$ through $P0_3$ to "L" and enter stop mode.

(2) If a key is pressed, "L" is input to one of pins $\overline{KI_0}$ through $\overline{KI_3}$ to clear stop mode. A key-input interrupt occurs to execute the key-input interrupt handling routine.

(3) Sequentially set $P0_0$ through $P0_3$ to "L" to determine which key was pressed.

(4) When the process to determine the key pressed is completed, change the output from $P0_0$ through $P0_3$ to "L" again and enter stop mode.

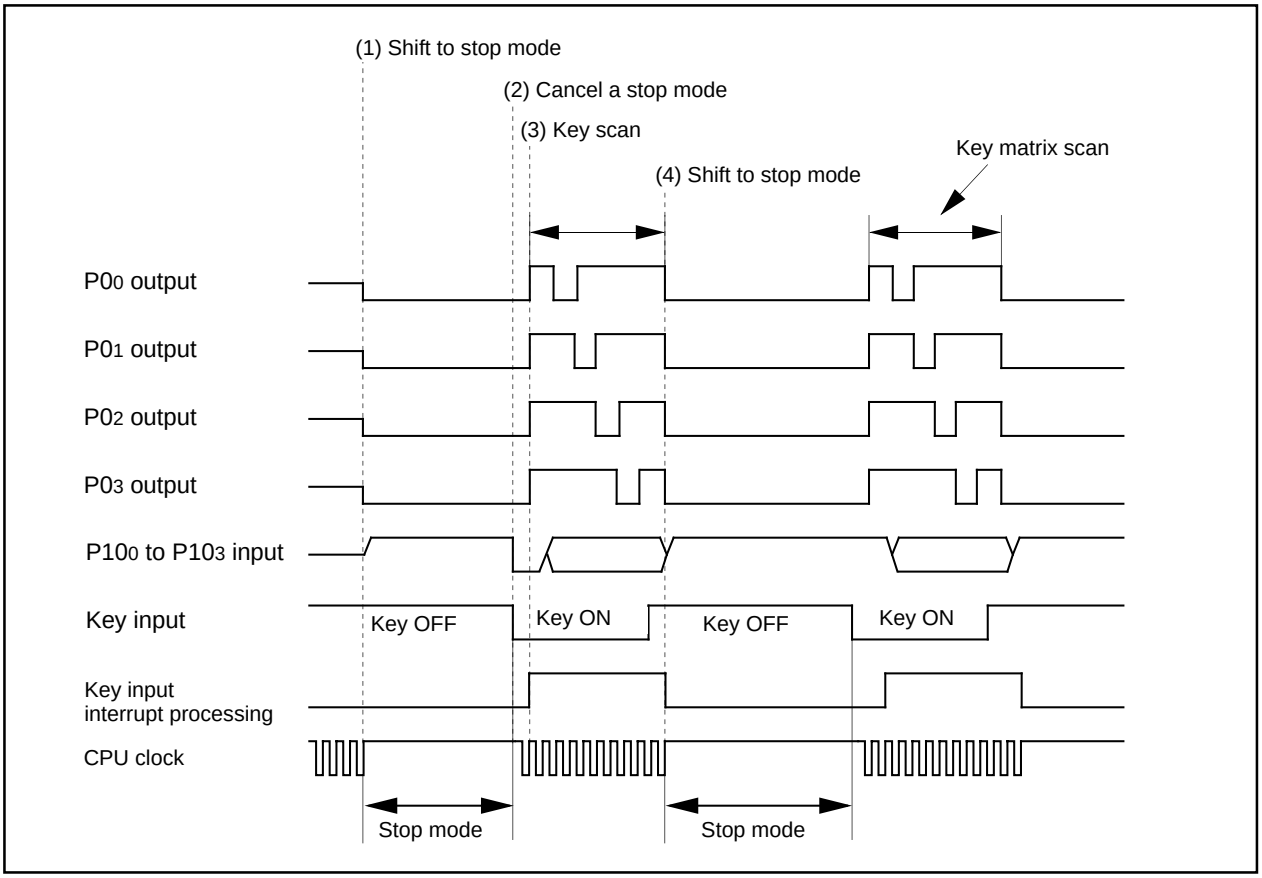


Figure 3.8.1. Operation timing of controlling power using stop mode

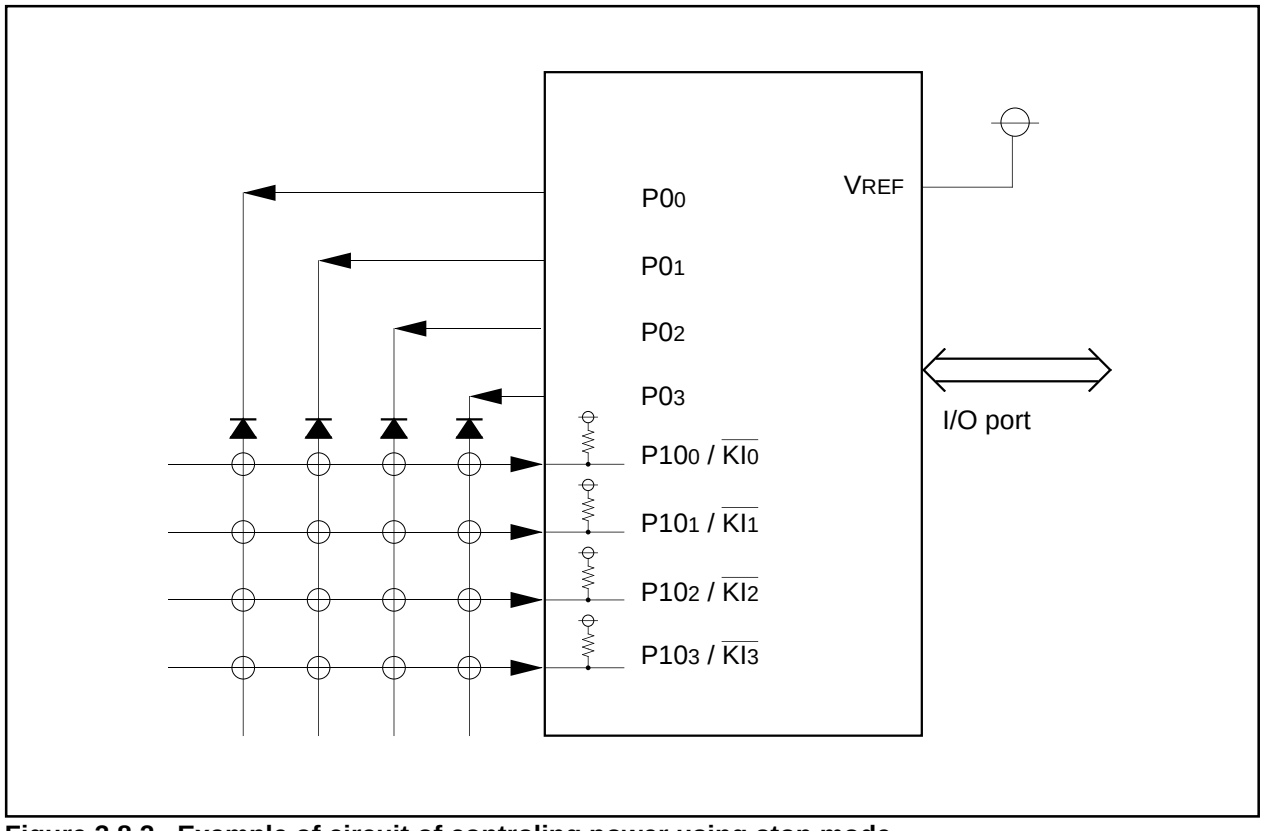


Figure 3.8.2. Example of circuit of controling power using stop mode

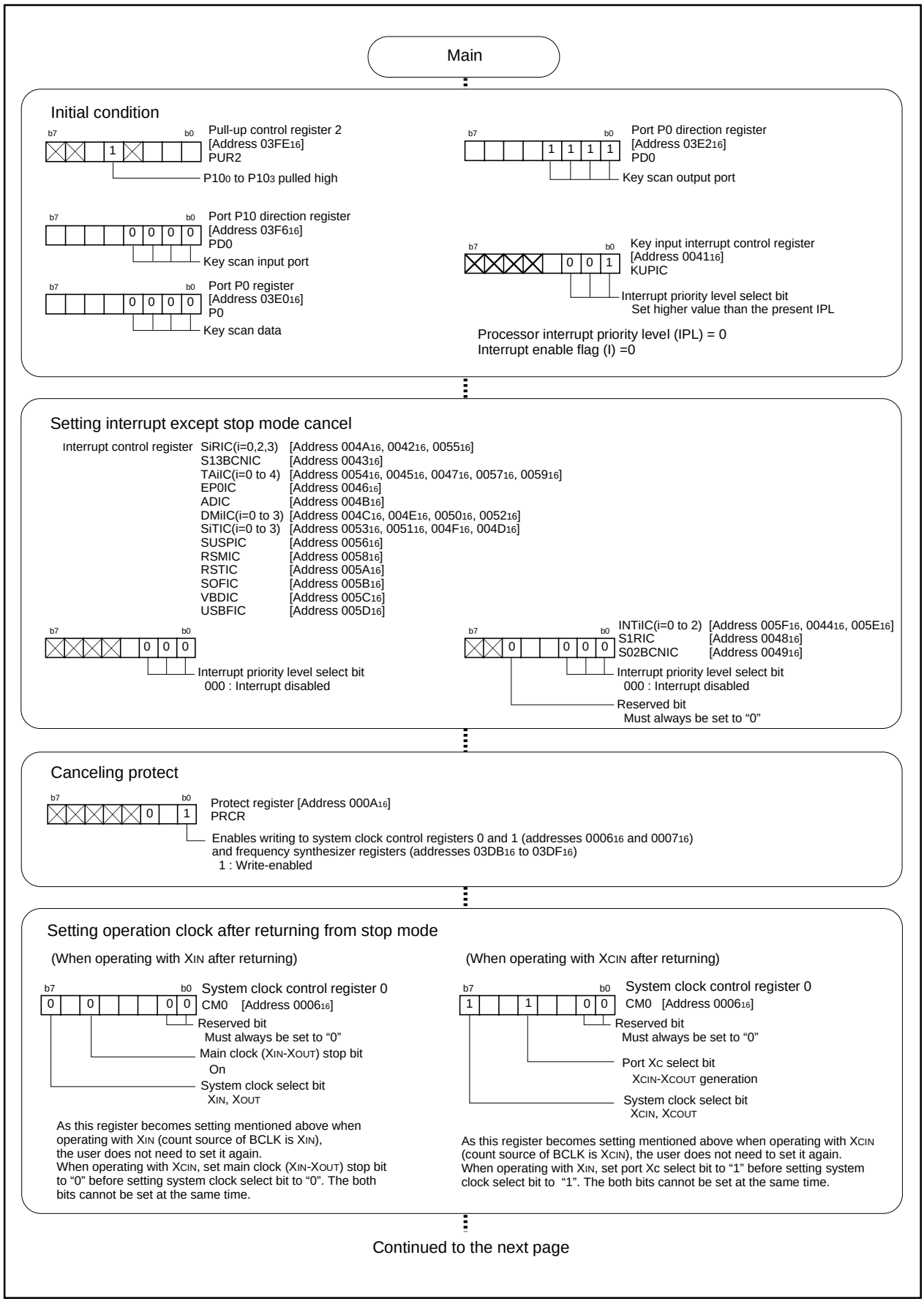

Main
Initial condition
b7 b0
1
Pull-up control register 2
[Address 03FE16]
PUR2
P100 to P103 pulled high
b7 b0
1 1 1 1
Port P0 direction register
[Address 03E216]
PD0
Key scan output port
b7 b0
0 0 0 0
Port P10 direction register
[Address 03F616]
PD0
Key scan input port
b7 b0
0 0 1
Key input interrupt control register
[Address 004116]
KUPIC
Interrupt priority level select bit
Set higher value than the present IPL
b7 b0
0 0 0 0
Port P0 register
[Address 03E016]
P0
Key scan data
Processor interrupt priority level (IPL) = 0
Interrupt enable flag (I) =0
Setting interrupt except stop mode cancel
Interrupt control register SiRIC(i=0,2,3) [Address 004A16, 004216, 005516]
S13BCNIC [Address 004316]
TAiIC(i=0 to 4) [Address 005416, 004516, 004716, 005716, 005916]
EP0IC [Address 004616]
ADIC [Address 004B16]
DMiIC(i=0 to 3) [Address 004C16, 004E16, 005016, 005216]
SiTIC(i=0 to 3) [Address 005316, 005116, 004F16, 004D16]
SUSPIC [Address 005616]
RSMIC [Address 005816]
RSTIC [Address 005A16]
SOFIC [Address 005B16]
VBDIC [Address 005C16]
USBFIC [Address 005D16]
b7 b0
0 0 0
Interrupt priority level select bit
000 : Interrupt disabled
b7 b0
0 0 0 0
INTiIC(i=0 to 2) [Address 005F16, 004416, 005E16]
S1RIC [Address 004816]
S02BCNIC [Address 004916]
Interrupt priority level select bit
000 : Interrupt disabled
Reserved bit
Must always be set to “0”
Canceling protect
b7 b0
0 1
Protect register [Address 000A16]
PRCR
Enables writing to system clock control registers 0 and 1 (addresses 000616 and 000716)
and frequency synthesizer registers (addresses 03DB16 to 03DF16)
1 : Write-enabled
Setting operation clock after returning from stop mode
(When operating with XIN after returning)
b7 b0
0 0 0 0
System clock control register 0
CM0 [Address 000616]
Reserved bit
Must always be set to “0”
Main clock (XIN-XOUT) stop bit
On
System clock select bit
XIN, XOUT
As this register becomes setting mentioned above when operating with XIN (count source of BCLK is XIN), the user does not need to set it again.
When operating with XCIN, set main clock (XIN-XOUT) stop bit to “0” before setting system clock select bit to “0”. The both bits cannot be set at the same time.
(When operating with XCIN after returning)
b7 b0
1 1 0 0
System clock control register 0
CM0 [Address 000616]
Reserved bit
Must always be set to “0”
Port XC select bit
XCIN-XCOUT generation
System clock select bit
XCIN, XCOUT
As this register becomes setting mentioned above when operating with XCIN (count source of BCLK is XCIN), the user does not need to set it again.
When operating with XIN, set port Xc select bit to “1” before setting system clock select bit to “1”. The both bits cannot be set at the same time.
Continued to the next page

Figure 3.8.3. Set-up procedure of controlling power using stop mode (1)

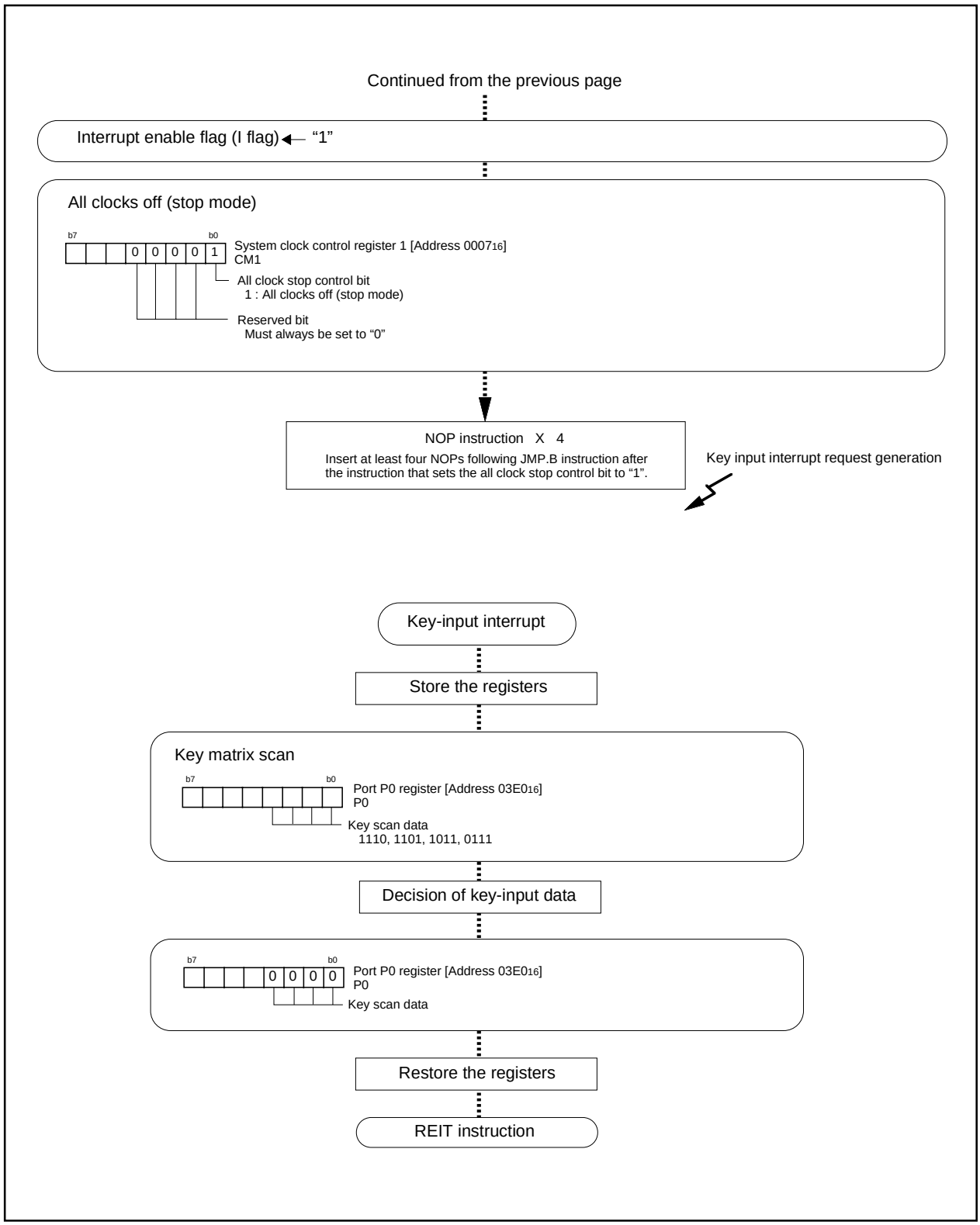

Continued from the previous page
Interrupt enable flag (I flag) ← "1"
All clocks off (stop mode)
b7 b0
0 0 0 0 1
System clock control register 1 [Address 0007₁₆]
CM1
All clock stop control bit
1 : All clocks off (stop mode)
Reserved bit
Must always be set to "0"
NOP instruction X 4
Insert at least four NOPs following JMP.B instruction after the instruction that sets the all clock stop control bit to "1".
Key input interrupt request generation
Key-input interrupt
Store the registers
Key matrix scan
b7 b0
Port P0 register [Address 03E0₁₆]
P0
Key scan data
1110, 1101, 1011, 0111
Decision of key-input data
b7 b0
0 0 0 0
Port P0 register [Address 03E0₁₆]
P0
Key scan data
Restore the registers
REIT instruction

Figure 3.8.4. Set-up procedure of controlling power using stop mode (2)

3.9 Controlling Power Using Wait Mode

Overview The following are steps for controling power using wait mode. Figure 3.9.1 shows the operation timing, and Figures 3.9.2 to 3.9.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- Wait mode

A flag named “F-WIT” is used in the set-up procedure. The purpose of this flag is to decide whether or not to clear wait mode. If F_WIT = “1” in the main program, the wait mode is entered; if F_WIT = “0”, the wait mode is cleared.

Specifications

(1) Connect a 32.768-kHz oscillator to XCIN to serve as the timer count source. As interrupts occur every one second, which is a count the timer reaches, the controller returns from wait mode and count the clock using a program.

(2) Clear wait mode if a $\overline{\text{INT0}}$ interrupt request occurs.

Operation (1) Switch the system clock from XIN to XCIN to get low-speed mode.

(2) Stop XIN and enter wait mode. In this instance, enable the timer A2 interrupt and the $\overline{\text{INT0}}$ interrupt.

(3) When a timer A2 interrupt request occurs (at 1-second intervals), start supplying the BCLK from XCIN. At this time, count the clock within the routine that handles the timer A2 interrupts and enter wait mode again.

(4) If a $\overline{\text{INT0}}$ interrupt occurs, start supplying the BCLK from XCIN. Start the XIN oscillation within the $\overline{\text{INT0}}$ interrupt, and switch the system clock to XIN.

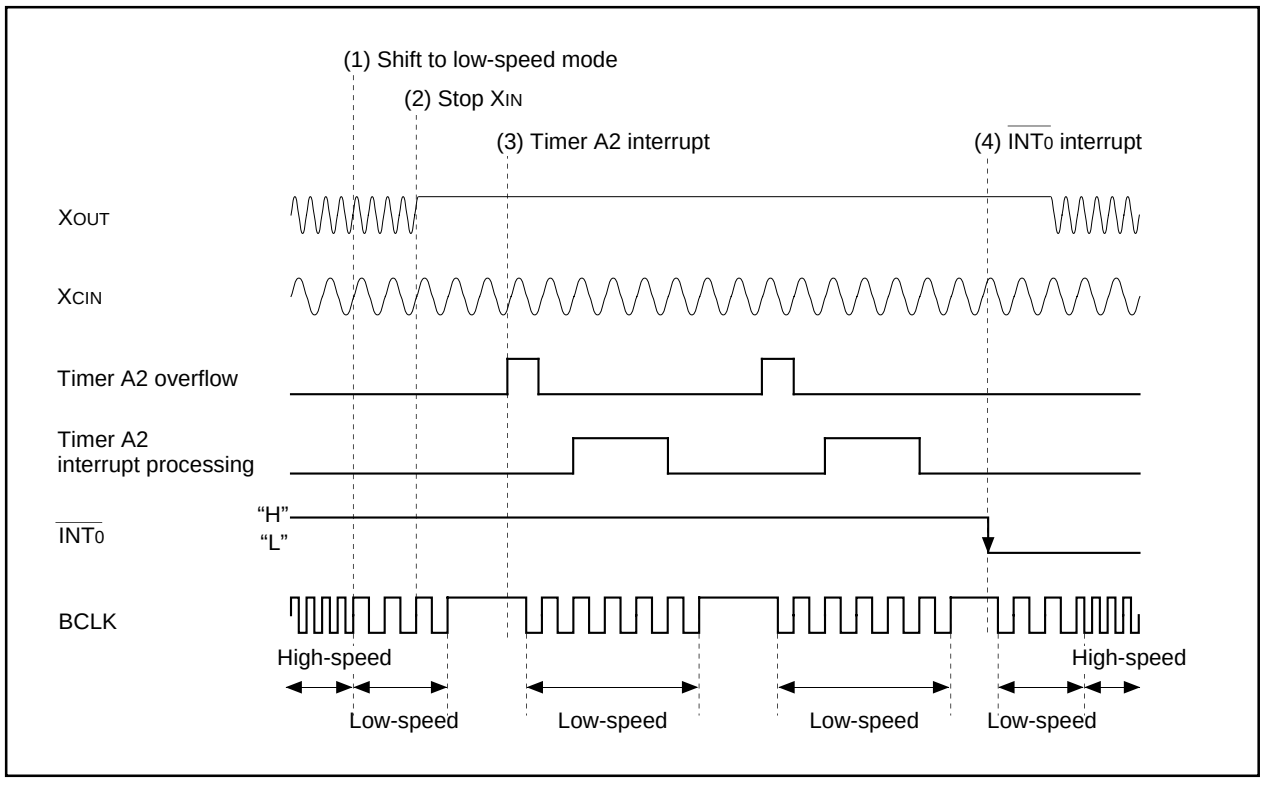


Figure 3.9.1. Operation timing of controling power using wait mode

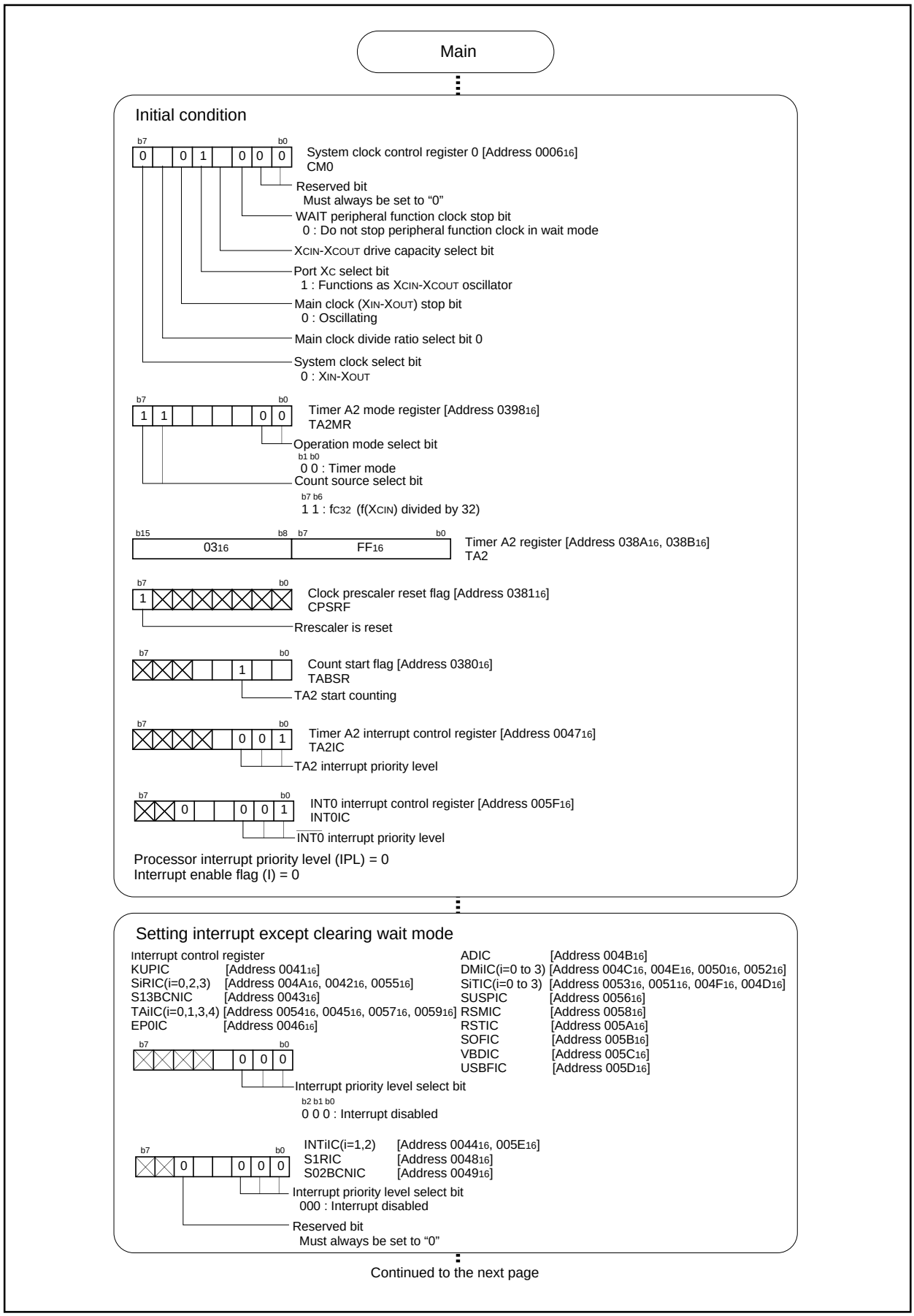


Figure 3.9.2. Set-up procedure of controlling power using wait mode (1)

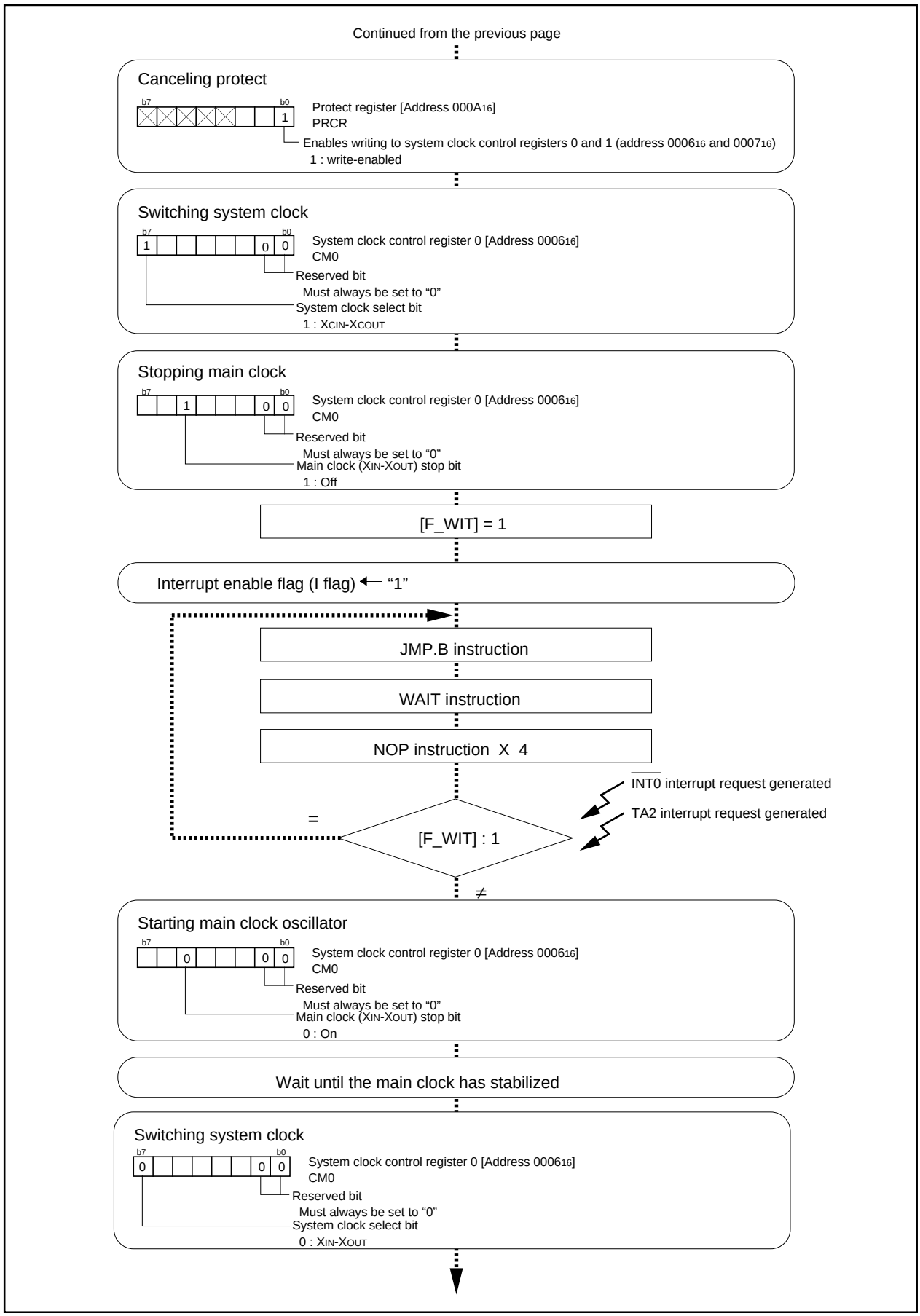


Figure 3.9.3. Set-up procedure of controlling power using wait mode (2)

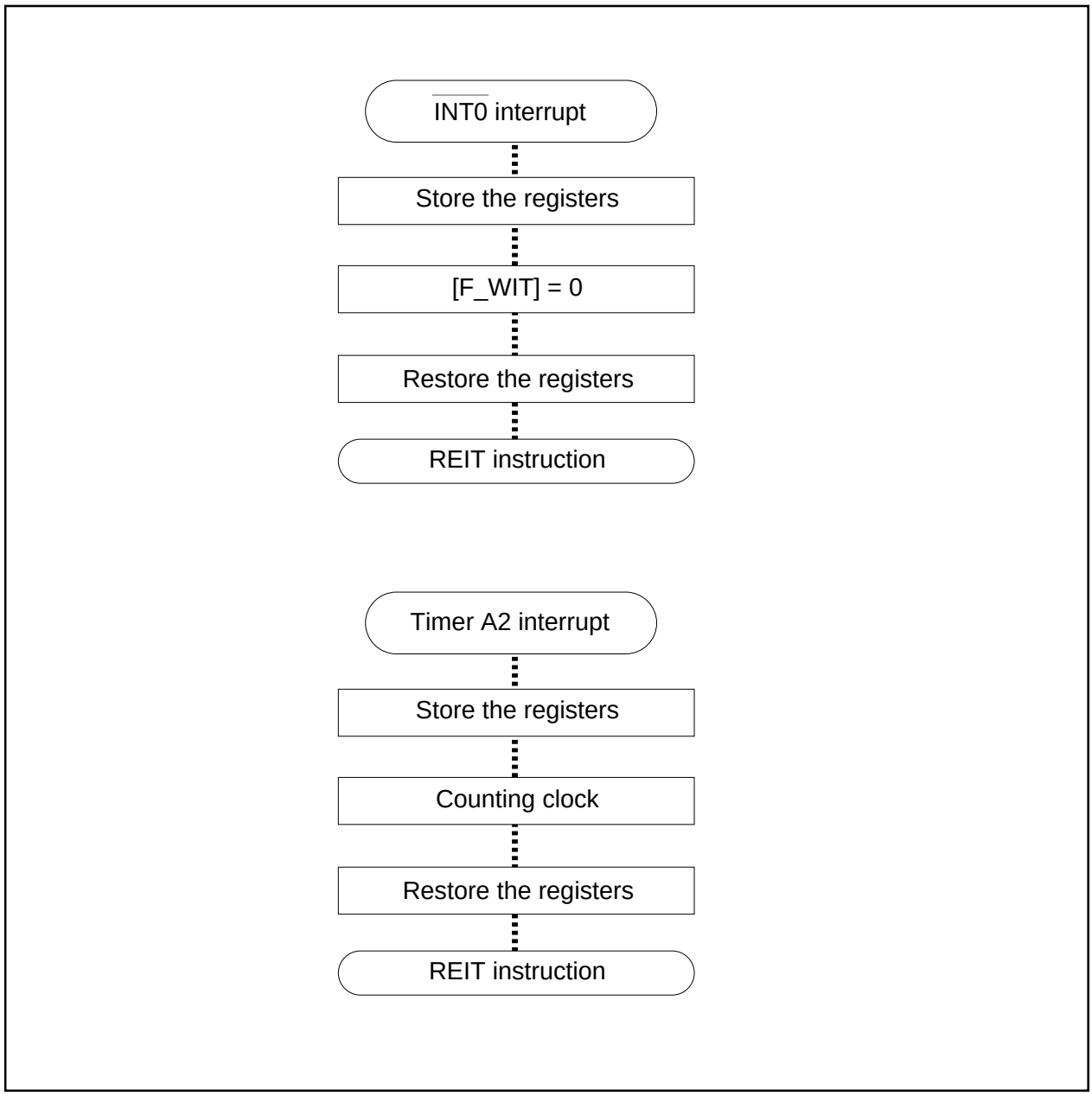


Figure 3.9.4. Set-up procedure of controlling power using wait mode (3)

Chapter 4

External Buses

4.1 Overview of External Buses

Memory and I/O external expansion can be connected to microcomputer easily by using external buses. When memory expansion mode or microprocessor mode is selected for processor mode, some of the pins function as the address bus, the data bus, and as control signals and this makes the external buses be able to operate.

When accessing an external area, 8-bit data bus width or 16-bit data bus width can be selected, based on the BYTE pin level. 16-bit width is used to access an internal area,regardless of the level of the BYTE pin. Fix the BYTE pin either to “H” or “L” level. 8-bit and 16-bit data bus widths cannot be used together in an external area.

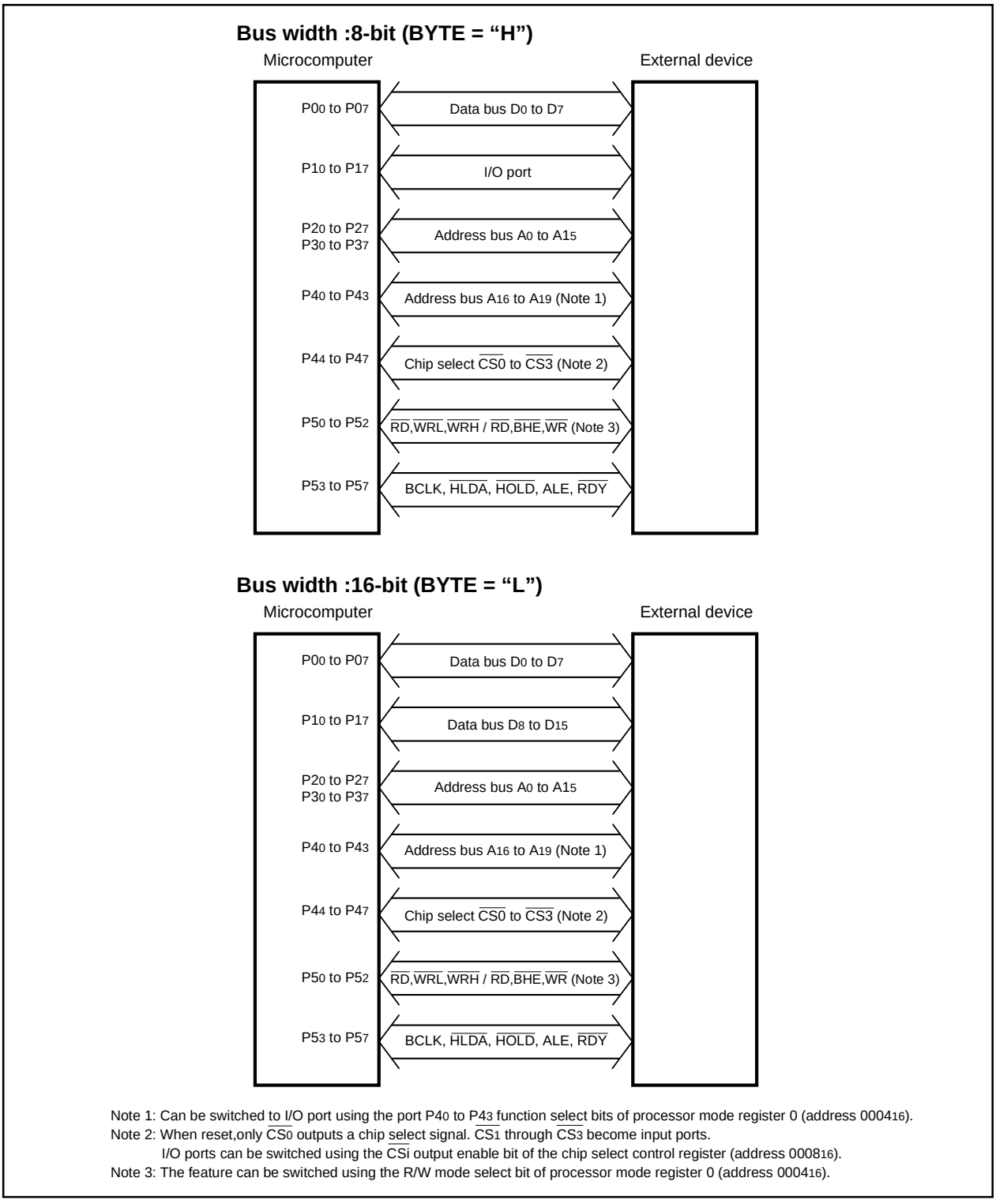
Make sure the BYTE pin is fixed to “H” level when an 8-bit bus width is selected and “L” level when a 16-bit bus width is selected.

4.2 Data Access

4.2.1 Data Bus Width

If the voltage level input to the BYTE pin is “H”, the external data bus width becomes 8 bits, and P1$_0$ (/D$_8$) through P1$_7$ (/D$_{15}$) can be used as I/O ports (Figure 4.2.1).

If the voltage level input to the BYTE pin is “L”, the external data bus width becomes 16 bits, and P0$_0$ (/D$_0$) through P0$_7$ (/D$_7$), and P1$_0$ (/D$_8$) through P1$_7$ (/D$_{15}$) operate as a data bus (D$_0$ through D$_{15}$) (Figure 4.2.1).



Note 1: Can be switched to I/O port using the port P4$_0$ to P4$_3$ function select bits of processor mode register 0 (address 0004$_{16}$).
Note 2: When reset,only CS$_0$ outputs a chip select signal. CS$_1$ through CS$_3$ become input ports.
I/O ports can be switched using the CSi output enable bit of the chip select control register (address 0008$_{16}$).
Note 3: The feature can be switched using the R/W mode select bit of processor mode register 0 (address 0004$_{16}$).

Figure 4.2.1. Level of BYTE pin and external data bus width

4.2.2 Chip Selects and Address Bus

Chip selects (P44/$\overline{CS0}$ through P47/$\overline{CS3}$) are output in areas resulting from dividing a 1-M byte memory space into four. To use the chip select, the chip select output must be enabled by setting the chip select control register. Figure 4.2.2 shows addresses in which chip selects become active (“L”). Since the extent of the internal area and the external area in memory expansion mode is different from those in microprocessor mode, there is a difference between areas for which $\overline{CS0}$ is output. When an internal ROM/RAM area is being accessed, no chip select is output, and the address bus does not change (the address of the external area that was accessed previously is held).

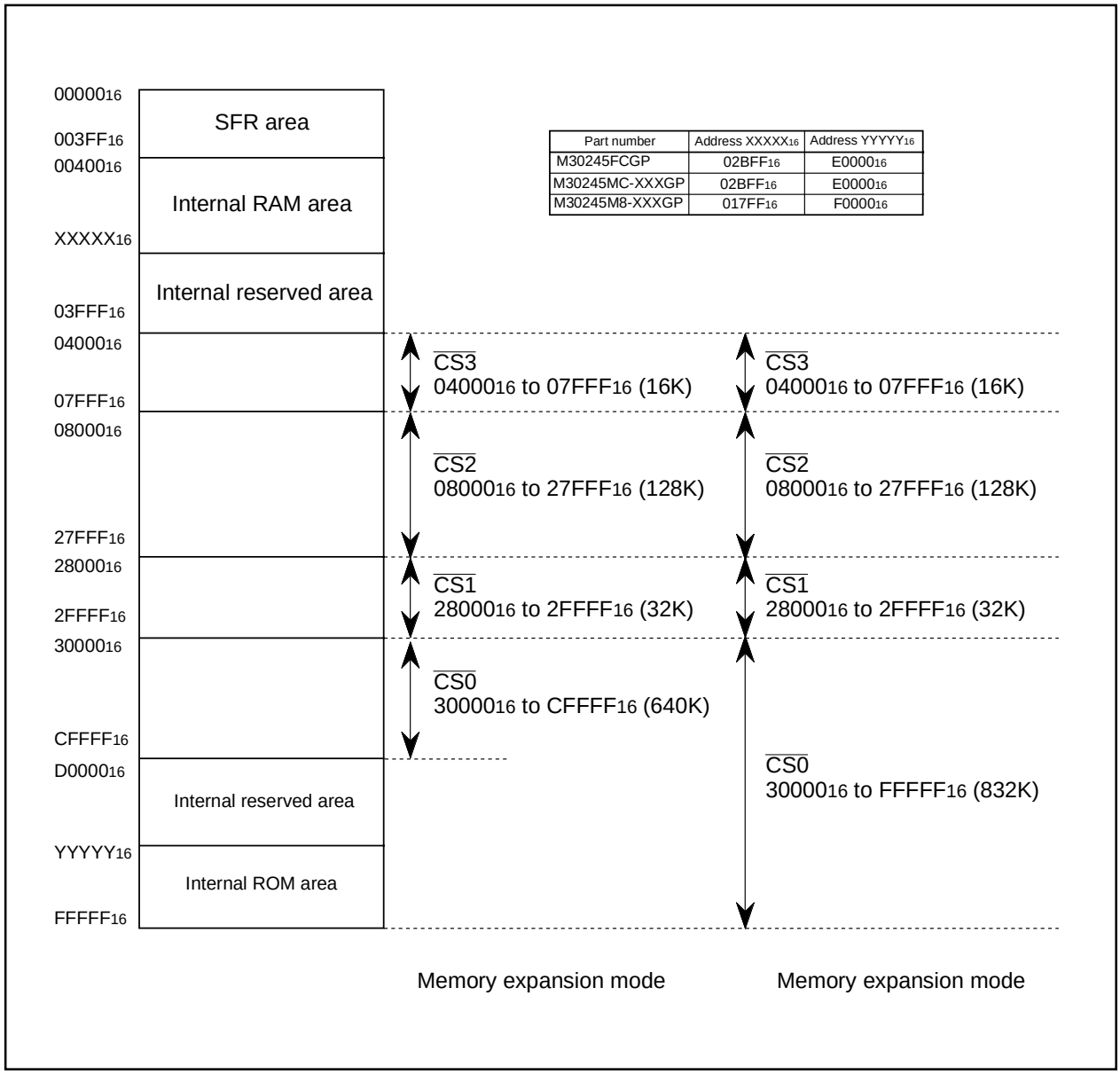


Figure 4.2.2. Addresses in which chip selects turn active (“L”)

Chip select control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CSR
Address: 0008₁₆
When reset: 01₁₆

Bit symbol	Bit name	Function	R	W
CS0	$\overline{CS0}$ output enable bit	0 : Chip select output disabled (Normal port pin) 1 : Chip select output enabled	○	○
CS1	$\overline{CS1}$ output enable bit		○	○
CS2	$\overline{CS2}$ output enable bit		○	○
CS3	$\overline{CS3}$ output enable bit		○	○
CS0W	$\overline{CS0}$ wait bit	0 : Wait state inserted 1 : No wait state	○	○
CS1W	$\overline{CS1}$ wait bit		○	○
CS2W	$\overline{CS2}$ wait bit		○	○
CS3W	$\overline{CS3}$ wait bit		○	○

Chip select expansion register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CSE
Address: 001B₁₆
When reset: 00₁₆

Bit symbol	Bit name	Function	R	W
CSE0W	$\overline{CS0}$ wait expansion bit	0 0 : 1 Wait state 0 1 : 2 Wait states 1 0 : 3 Wait states 1 1 : Inhibited	○	○
CSE1W	$\overline{CS1}$ wait expansion bit		○	○
CSE2W	$\overline{CS2}$ wait expansion bit		○	○
CSE3W	$\overline{CS3}$ wait expansion bit		○	○

Note 1: Set CSEiW bits (i = 0 to 3) after setting the corresponding CSiW bit (i = 0 to 3) of the CSR register to “0”. When CSiW bits are set to “1”, CSEiW bits must be returned to “00₂”.

Figure 4.2.3. Level of BYTE pin and external data bus width

4.2.3 R/W Modes

The read/write signal that is output when accessing an external area can be selected between the $\overline{RD}$/$\overline{BHE}$/$\overline{WR}$ and the $\overline{RD}$/$\overline{WRH}$/$\overline{WRL}$ modes by setting the R/W mode select bit (bit 2) of the processor mode register 0 (address 0004₁₆). Use the ($\overline{RD}$/$\overline{BHE}$/$\overline{WR}$) mode to access an 8-bit and a 16-bit wide RAM, and the ($\overline{RD}$/$\overline{WRH}$/$\overline{WRL}$) to access a 16-bit wide RAM.

When the M30245 is reset, the $\overline{RD}$/$\overline{BHE}$/$\overline{WR}$ mode is selected by default. To switch over the R/W mode, change the $\overline{RD}$/$\overline{BHE}$/$\overline{WR}$ to the $\overline{RD}$/$\overline{WRH}$/$\overline{WRL}$ mode before accessing an external RAM.

Refer to the connection examples of $\overline{RD}$/$\overline{BHE}$/$\overline{WR}$ and $\overline{RD}$/$\overline{WRH}$/$\overline{WRL}$ shown in Section 4.3, “Connection Examples.”

4.3 Connection Examples

4.3.1 16-bit Memory to 16-bit Width Data Bus Connection Example

Figure 4.3.1 shows an example of connecting M5M51016BTP (SRAM). In this diagram, when reset the microcomputer starts operating in single-chip mode. Change this mode to memory expansion mode in a program.

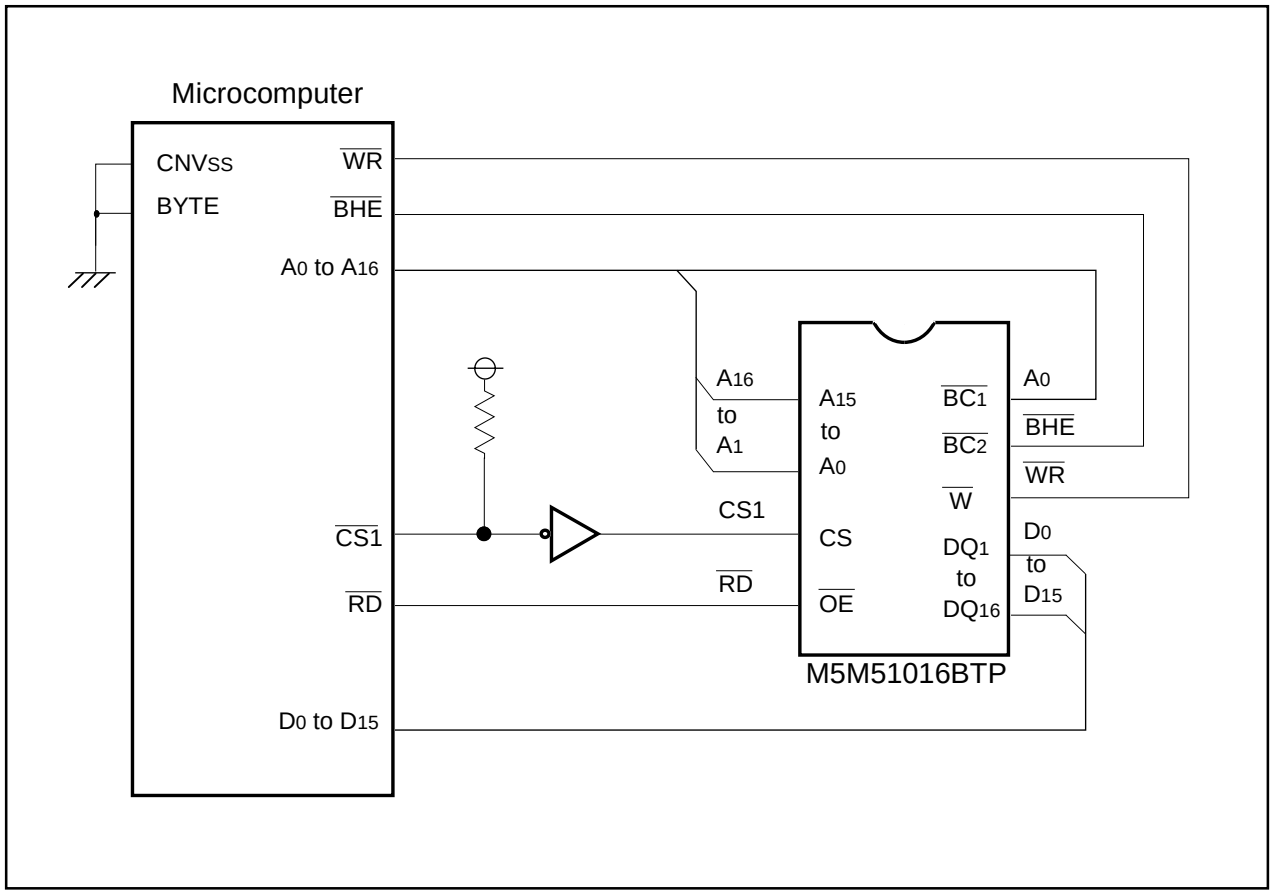


Figure 4.3.1. Example of connecting M5M51016BTP

4.3.2 8-bit Memory to 16-bit Width Data Bus Connection Example

Figure 4.3.2 shows an example of connecting two M5M5278's (SRAM) to a 16-bit data bus.In this diagram, when reset the microcomputer starts operating in single-chip mode. Change this mode to memory expansion mode in a program.

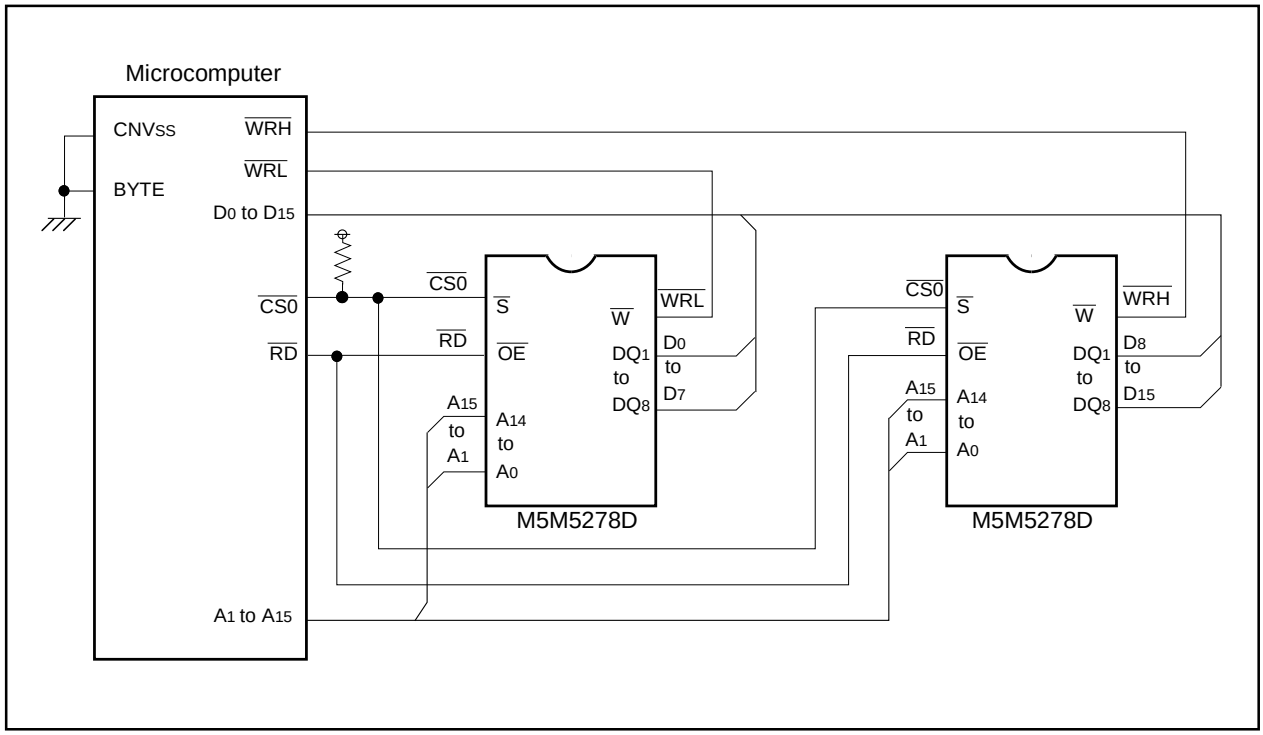


Figure 4.3.2. Example of connecting two M5M5278's to a 16-bit data bus

Figure 4.3.3 shows how to connect two Am29LV008B (flash memory). In 16-bit bus mode,the $\overline{BHE}$/$\overline{WRH}$ pin functions as $\overline{BHE}$. When connecting 8-bit flash memory chips to the 16-bit bus, make sure the microcomputer's $\overline{WRL}$ pin is connected to the $\overline{WR}$ pins on both flash memory chips, and that data is written to the flash memory in units of 16 bits beginning with an even address.

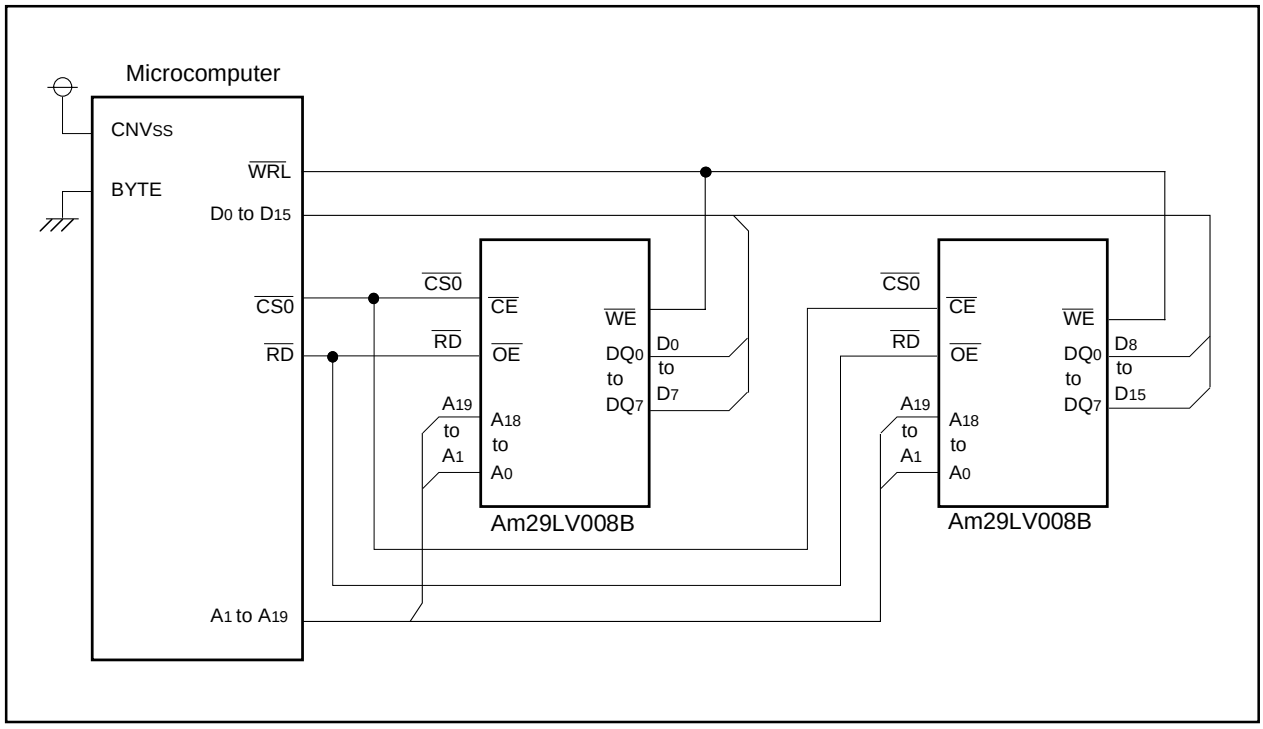


Figure 4.3.3. Example of connecting two Am29LV008B's to a 16-bit data bus

4.3.3 8-bit Memory to 8-bit Width Data Bus Connection Example

Figure 4.3.4 shows an example of connecting two M5M5278's (SRAM) to an 8-bit data bus.In this diagram, when reset the microcomputer starts operating in single-chip mode. Change this mode to memory expansion mode in a program.

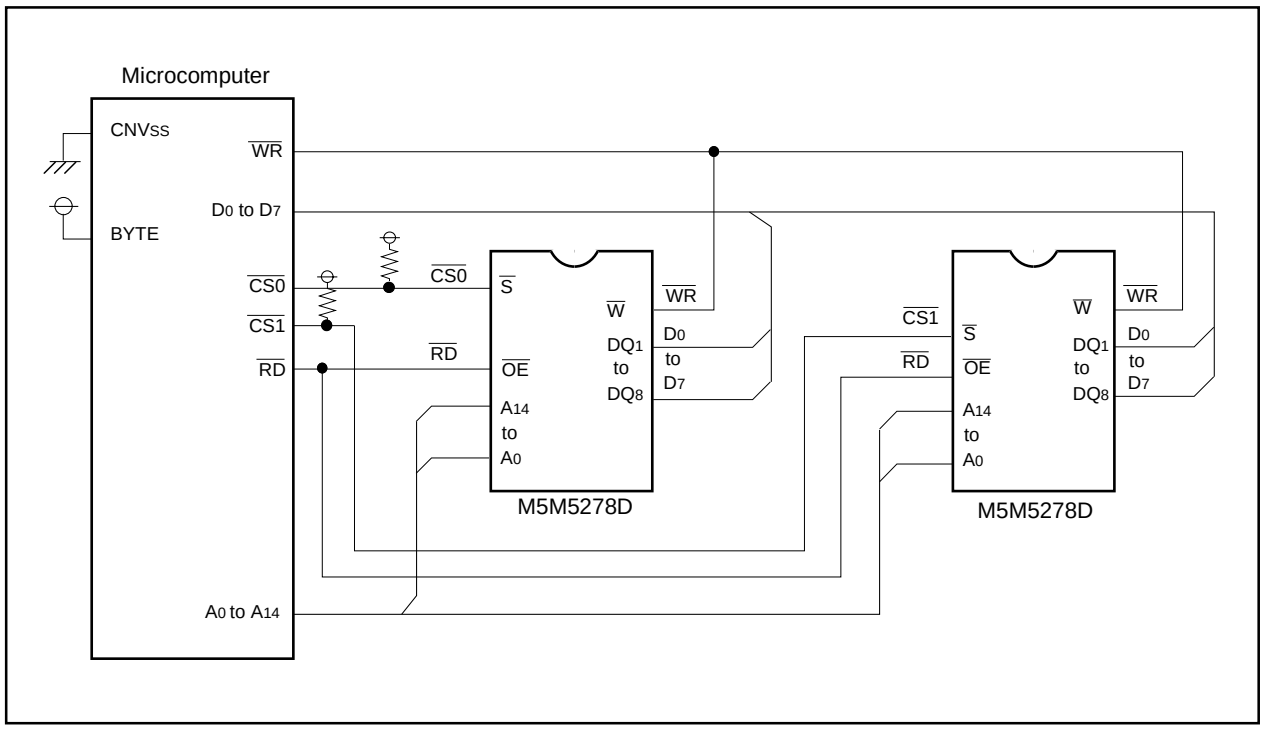


Figure 4.3.4. Example of connecting two M5M5278's to an 8-bit data bus

4.3.4 Two 8-bit and 16-Bit Memory to 16-Bit Width Data Bus Connection Example

Figure 4.3.5 shows an example of connecting M5M28F102 (16-bit flash memory) and two M5M5278's (8-bit SRAM) to a 16-bit data bus.

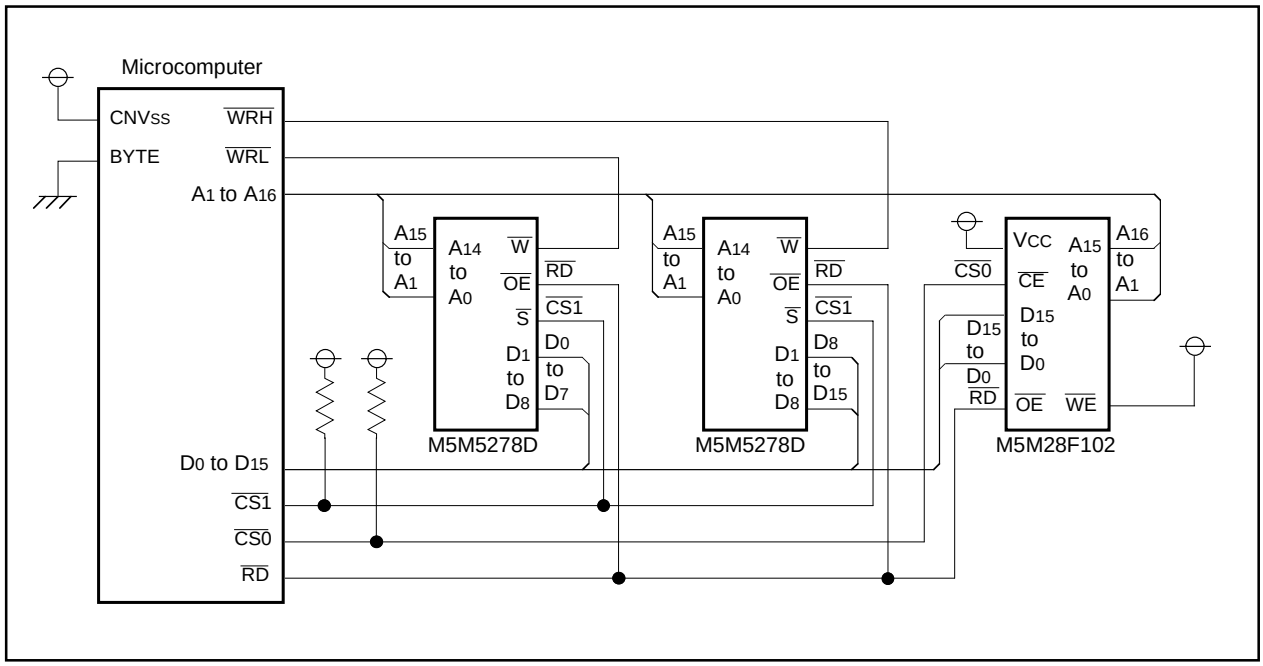


Figure 4.3.5. Example of connection of two 8-bit memories and one 16-bit memory to 16-bit width data bus

4.3.5 Chip Selects and Address Bus

When there are insufficient chip select signals, it is necessary to generate chip selects externally. Figure 4.3.6 shows an example of a connection in which the $\overline{CS2}$ (128K bytes) area is divided into four 32K byte areas.

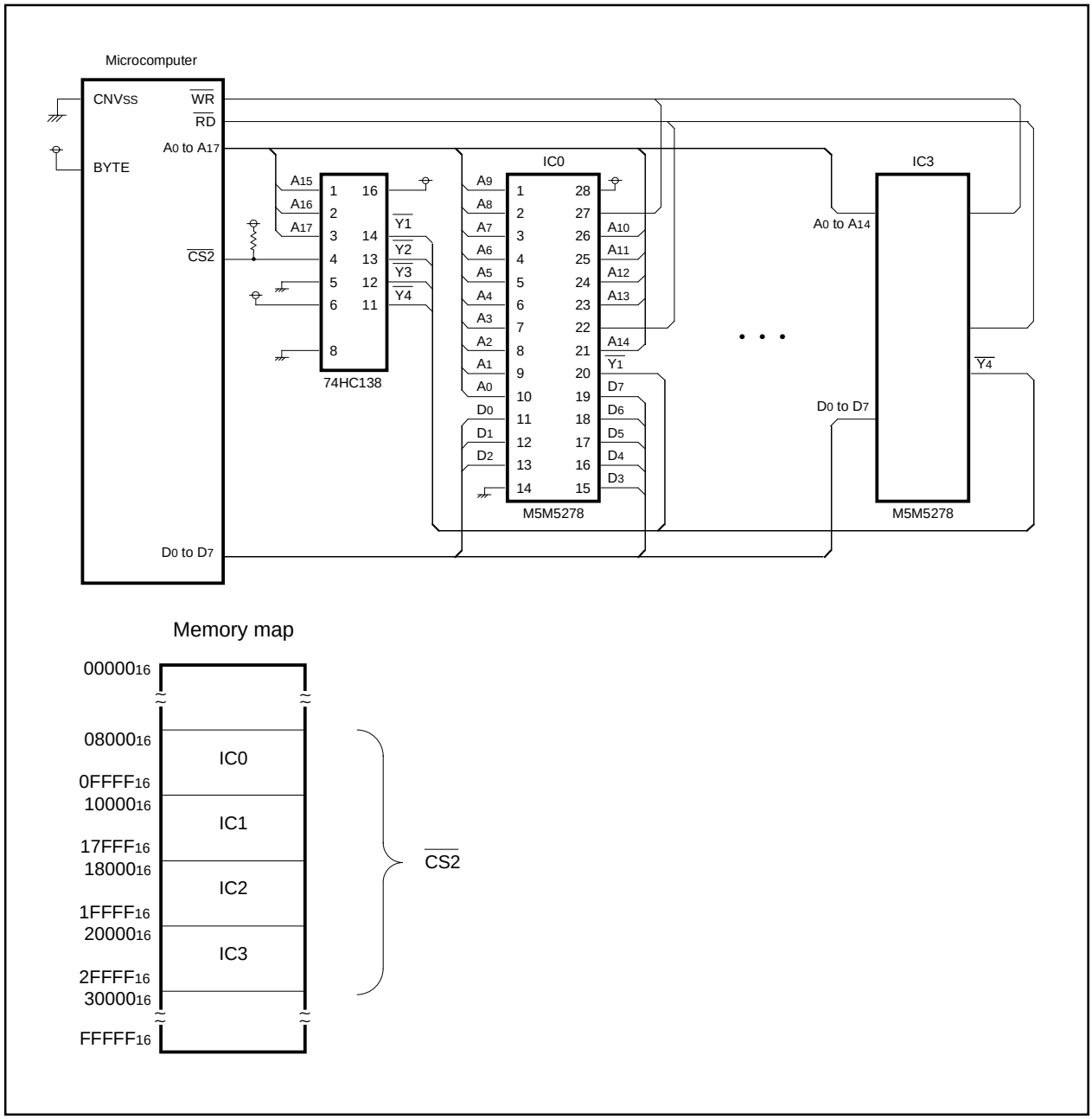


Figure 4.3.6. Chip selects and address bus

4.4 Connectable Memories

4.4.1 Operation Frequency and Access Time

Connectable memories depend upon the BCLK frequency f(BCLK). The frequency of f(BCLK) is equal to that of the BCLK, and is contingent on the oscillator's frequency and on the settings in the system clock select bits (bit 6 of address 0006_{16}, and bits 6 and 7 of address 0007_{16}).

The following are the conditional equations for the connections. Meet these conditions minimally. Figures 4.4.1 and 4.4.2 show the relation between the frequency of BCLK and memory.

(1) Read cycle time (tCR)/write cycle time (tCW)

Read cycle time (tCR) and write cycle time (tCW) must satisfy the following conditional expressions:

- **With the Wait option cleared**

  $tCR < 10^9/f(BCLK)$ and $tCW < 2 \times 10^9/f(BCLK)$

  (When CSxW = 1 read: one cycle of BCLK write: two cycles of BCLK)

- **With the Wait option selected**

  $tCR < (m+1) \times 10^9/f(BCLK)$ and $tCW < (m+1) \times 10^9/f(BCLK)$

  (When CSxW = 0 and the number of the expansion waits is selected by the CSExW bit)

  (m denotes the number of Wait states: m = “1” when 1 wait selected, “m = 2” when 2 waits selected, and “m = 3” when 3 waits selected)

(2) Address access time [ta(A)]

Address access time [ta(A)] must satisfy the following conditional expressions:

(a) Vcc = 3.0 to 3.6 V

- **With the Wait option cleared**

  $ta(A) < 10^9/f(BCLK) – 80(ns)$*

- **With the Wait option selected**

  $ta(A) < (m+1) \times 10^9/f(BCLK) – 80(ns)$*

  (m = “1” when 1 wait selected, “m = 2” when 2 waits selected, and “m = 3” when 3 waits selected)

  *80(ns) = td(BCLK – AD) + tsu(DB – RD) – th(BCLK – RD)

  = (address output delay time) + (data input setup time) – (RD signal output hold time)

(3) Chip select access time [ta(S)]

Chip select access time [ta(S)] must satisfy the following conditional expressions:

(a) Vcc = 3.0 to 3.6 V

- **With the Wait option cleared**

  $ta(S) < 10^9/f(BCLK) – 80(ns)$*

- **With the Wait option selected**

  $ta(S) < (m+1) \times 10^9/f(BCLK) – 80(ns)$*

  (m = “1” when 1 wait selected, “m = 2” when 2 waits selected, and “m = 3” when 3 waits selected)

  *80(ns) = td(BCLK – CS) + tsu(DB – RD) – th(BCLK – RD)

  = (chip select output delay time) + (data input setup time) – (RD signal output hold time)

(4) Output enable time [ta(OE)]

Output enable time [ta(OE)] must satisfy the following conditional expressions:

(a) Vcc = 3.0 to 3.6 V

• With the Wait option cleared

ta(OE) < 10^9/(f(BCLK) × 2) – 60(ns) = tac1(RD-DB)

• With the Wait option selected

ta(OE) < (m+0.5) × 10^9/f(BCLK) – 60(ns) = tac2(RD-DB)

(m = "1" when 1 wait selected, "m = 2" when 2 waits selected, and "m = 3" when 3 waits selected)

(5) Data setup time [tsu(D)]

Data setup time [tsu(D)] must satisfy the following conditional expressions:

(a) Vcc = 3.0 to 3.6 V

• PM16 = 0 (WR width normal)

tsu(D) < (n – 0.5) × 10^9/f(BCLK) – 40(ns) = td(DB – WR)

• PM16 = 1 (WR width expanded)

tsu(D) < n × 10^9/f(BCLK) – 40((ns) = td(DB – WR)

*40(ns) = td(BCLK – DB) – th(BCLK – WR)

= (data output delay time) – (WR signal output hold time)

(n = 1 (no wait), n = 2 (1 wait), n = 3 (2 waits), n = 4 (3 waits))

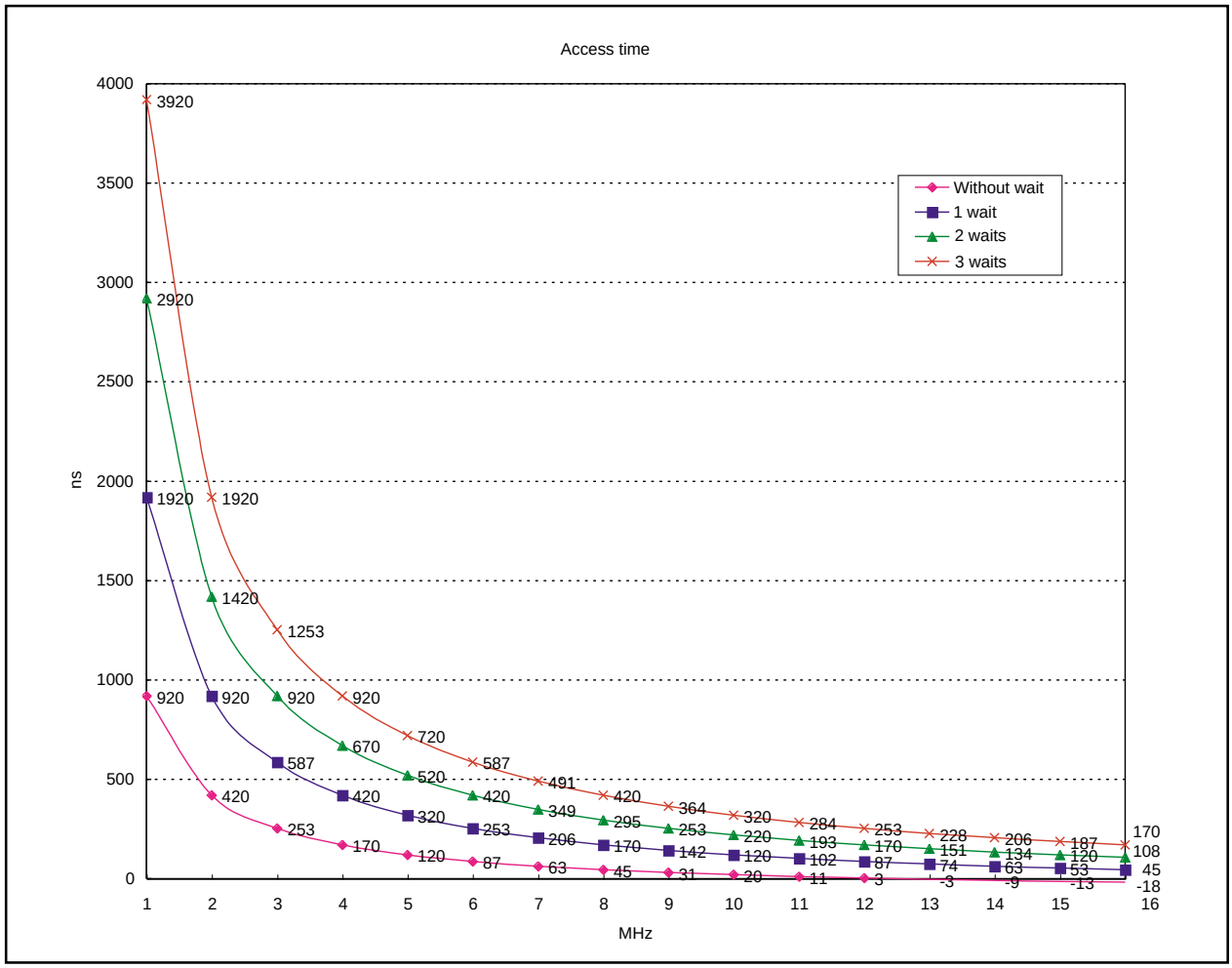


Figure 4.4.1. Relation between the frequency of BCLK and memory (1)

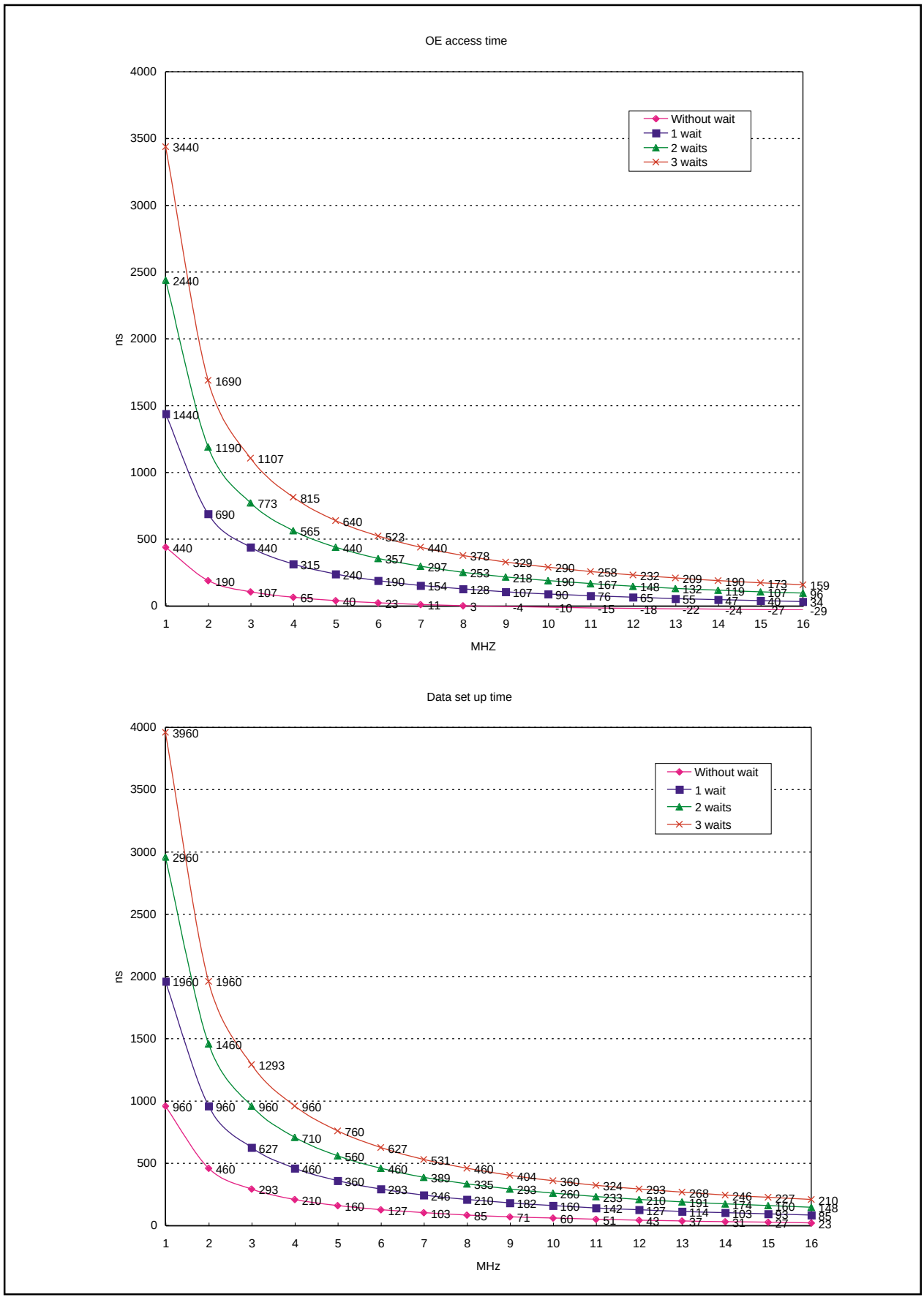


Figure 4.4.1. Relation between the frequency of BCLK and memory (2)

4.4.2 Connecting Low-Speed Memory

To connect memory with long access time [ta(A)], either decrease the frequency of BCLK or set a software wait. Using the $\overline{RDY}$ feature allows you to connect memory having the timing that precludes connection though you set software wait.

(1) Using software wait

Set software wait for the external memory areas by using either of bits CS0W through CS3W of the chip select control register (address 0008_{16}) or the chip select expansion register (address $001B_{16}$). When using the $\overline{RDY}$ signal, the corresponding CS0W through CS3W bit must be set to "0".

Bits CS0W through CS3W of the chip select control register correspond to chip select CS0 through CS3, respectively. If these bits are set to "1", the read bus cycle results in one cycle of BCLK and the write bus cycle results in two cycles of BCLK; if these bits are set to "0", the read/write cycle results in two, three, or four cycles of BCLK according to the chip select expansion register setting. When the corresponding bit of the chip select control register is set to "0", the chip select expansion register setting becomes valid. When this bit is set to "1", set the corresponding bit of the chip select expansion register to "00_2".

When reset, the value of the chip select control register and the chip select expansion register is set to "00_{16}".

These control bits do not affect the SFR area and the internal ROM/RAM area.

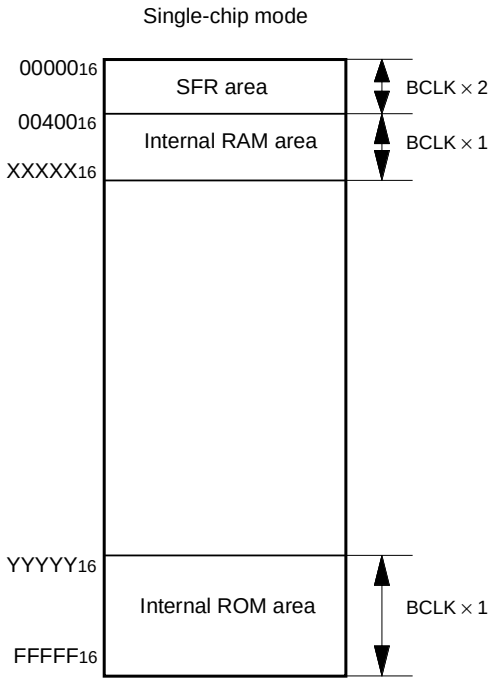
Figure 4.4.1 shows software wait and bus cycle, and Figure 4.4.3 shows relation of processor mode and the wait bit (CSiW, CSiEW).

Table 4.4.1. Software waits and bus cycles

Area	CSxW (Note 2)	CSExW (Note 2)	Bus Cycles	
			Read	Write
SFR	Invalid	Invalid	2 BCLK cycles	2 BCLK cycles
Internal ROM/RAM	Invalid	Invalid	1 BCLK cycle	1 BCLK cycle
External memory areas	0	00	2 BCLK cycles	2 BCLK cycles
	0	01	3 BCLK cycles	3 BCLK cycles
	0	10	4 BCLK cycles	4 BCLK cycles
	0	11	Do not set	
	1	00	1 BCLK cycle	2 BCLK cycles

Note 1: When using the $\overline{RDY}$ signal, set to "0".

Note 2: Set CSEiW bits (i = 0 to 3) after setting the corresponding CSiW bit (i = 0 to 3) of the CSR register to "0". When CSiW bits are set to "1", CSEiW bits must be returned to "00_2".



Type No.	Address XXXXX16	Address YYYYY16
M30245FCGP	02BFF16	E000016
M30245MC-XXXGP	02BFF16	E000016
M30245M8-XXXGP	017FF16	F000016

- Memory expansion mode and microprocessor examples apply with the following settings:

CS0 = “1”($\overline{CS0}$ output enabled)
CS3 = “1”($\overline{CS3}$ output enabled)
CS0W = “0” (with $\overline{CS0}$ wait)
CS3W = “1” (without $\overline{CS3}$ wait)
CSE0W = “012” ($\overline{CS0}$: 2-wait expansion)

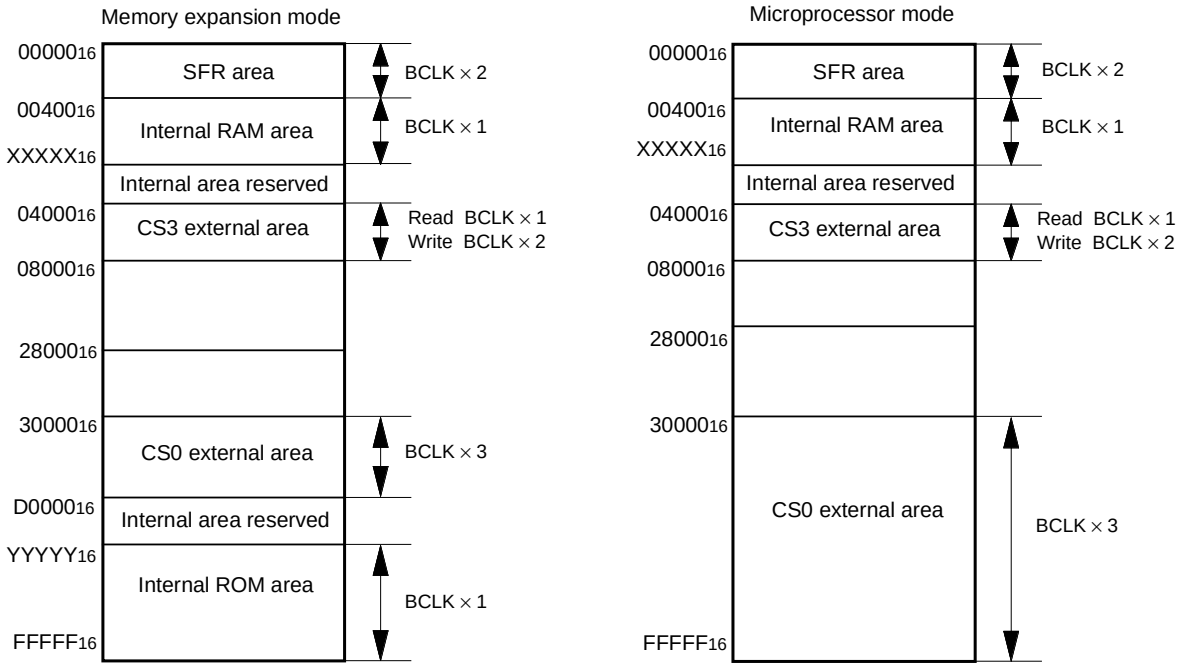


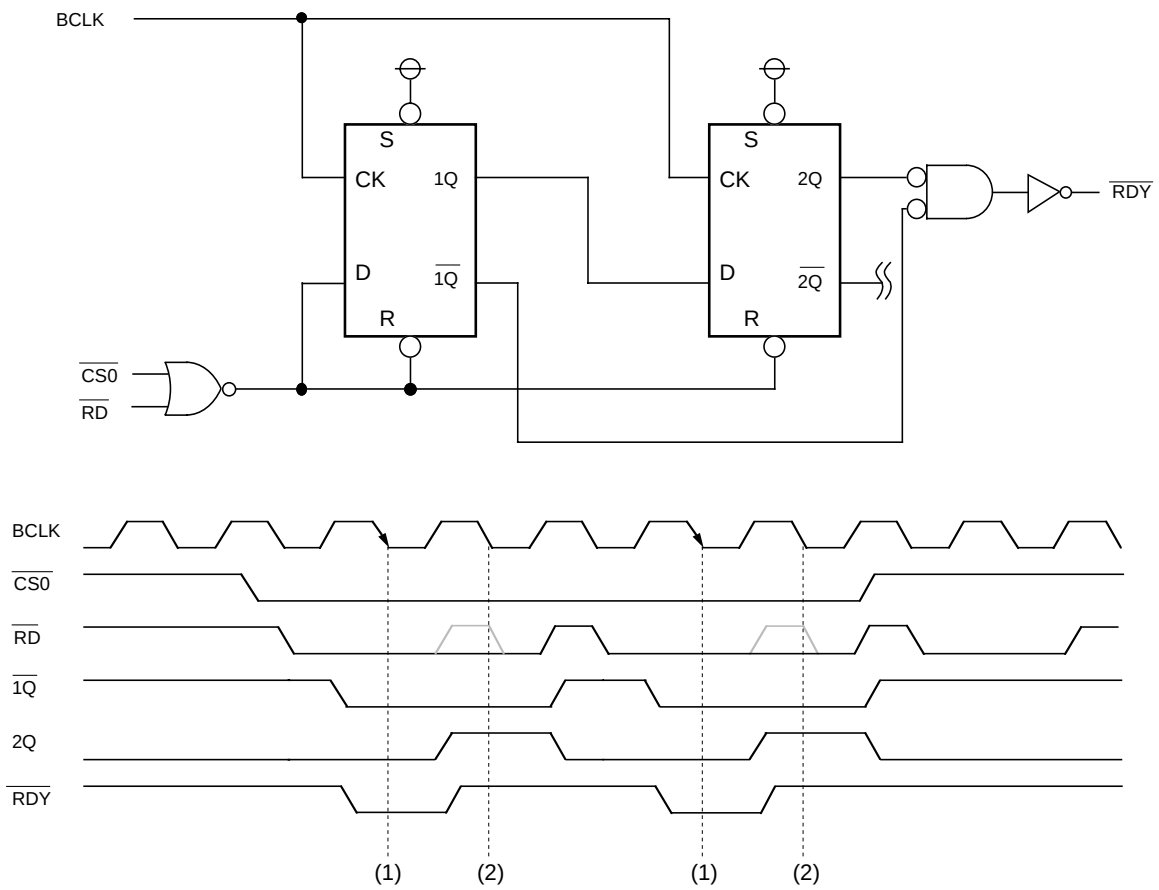
Figure 4.4.3. Relation of processor mode and the wait bit (CSiW, CSiEW)

(2) $\overline{\text{RDY}}$ function usage

To use the $\overline{\text{RDY}}$ function, set a software wait.

The $\overline{\text{RDY}}$ function operates when the BCLK signal falls with the $\overline{\text{RDY}}$ pin at “L”; the bus does not vary for 1 BCLK, and the state at that moment is held.

The $\overline{\text{RDY}}$ function holds the state of bus for the period in which the $\overline{\text{RDY}}$ pin is at “L”, and releases it when the BCLK signal falls with the $\overline{\text{RDY}}$ pin at “H”. Figure 4.4.4 shows an example of $\overline{\text{RDY}}$ circuit (f(XIN)=10MHz) that holds the state of bus for 1 BCLK.



(1) $\overline{\text{RDY}}$ accepted
(2) $\overline{\text{RDY}}$ cleared
The state of data bus and that of address bus are held for the period between (1) and (2).

Figure 4.4.4. Example of $\overline{\text{RDY}}$ circuit holding state of bus for 1 BCLK (f(XIN)=10MHz)

4.4.3 Connectable Memories

Connectable memories and their maximum frequencies are given here;
M30245 group maximum frequency is
16MHz (without the wait) for Vcc=3V,

(1) Flash memories (Read only mode)

(a) 3V without wait

Maximum frequency (MHz)	Model No.
3.57	M5M29GB/T160BVP-80

(b) 3V with wait

Maximum frequency (MHz)	Model No.
8.33	M5M29GB/T160BVP-80

(2) SRAM

(a) 3V without wait

Maximum frequency (MHz)	Model No.
5.12	M5M54R08AJ-12 M5M54R16AJ, ATP-12

(b) 3V without wait

Maximum frequency (MHz)	Model No.
10.0	M5M54R08AJ-12 M5M54R16AJ, ATP-12

4.5 Releasing an External Bus ($\overline{\text{HOLD}}$ input and $\overline{\text{HLDA}}$ output)

The Hold feature is to relinquish the address bus, the data bus, and the control bus on M30245 side in line with the Hold request from the bus master other than M30245 when the two or more bus masters share the address bus, the data bus, and the control bus. The Hold feature is effective only in memory expansion mode and microprocessor mode.

The sequence of using the Hold feature may be:

1. The external bus master turns the input level of the $\overline{\text{HOLD}}$ terminal to “L”.
2. When M30245 becomes ready to relinquish buses, each bus becomes high-impedance state at the falling edge of BCLK.
3. The $\overline{\text{HLDA}}$ terminal becomes “L” at the rising edge of the next BCLK.
4. The external bus master uses a bus.
5. When the external bus master finishes using a bus, the external bus master returns the input level of the $\overline{\text{HOLD}}$ terminal to “H”.
6. The output from $\overline{\text{HLDA}}$ terminal becomes “H” at the rising edge of the next BCLK.
7. Each bus returns from the high-impedance state to the former state at the falling edge of the next BCLK.

As given above, each bus invariably gets in the high-impedance state while the $\overline{\text{HLDA}}$ output is “L”. Also, M30245 does not relinquish buses during a bus cycle. That is, if a Hold request comes in during a bus cycle, the $\overline{\text{HLDA}}$ output become “L” after that bus cycle finishes.

In the Hold state, the state of each terminal becomes as follows.

- **Address bus A_0 to A_{19}**

  High-impedance state. The case in which A_{16} to A_{19} are used as ports $P4_0$ to $P4_3$ (64K byte address space) in microprocessor mode and in memory expansion mode too falls under this category.

- **Data bus D_0 to D_{15}**

  High-impedance state. The case in which D_8 to D_{15} are used as ports $P1_0$ to $P1_7$ (8-bit external bus width) in microprocessor mode and in memory expansion mode too falls under this category.

- **$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{BHE}}$**

  High-impedance state.

- **ALE**

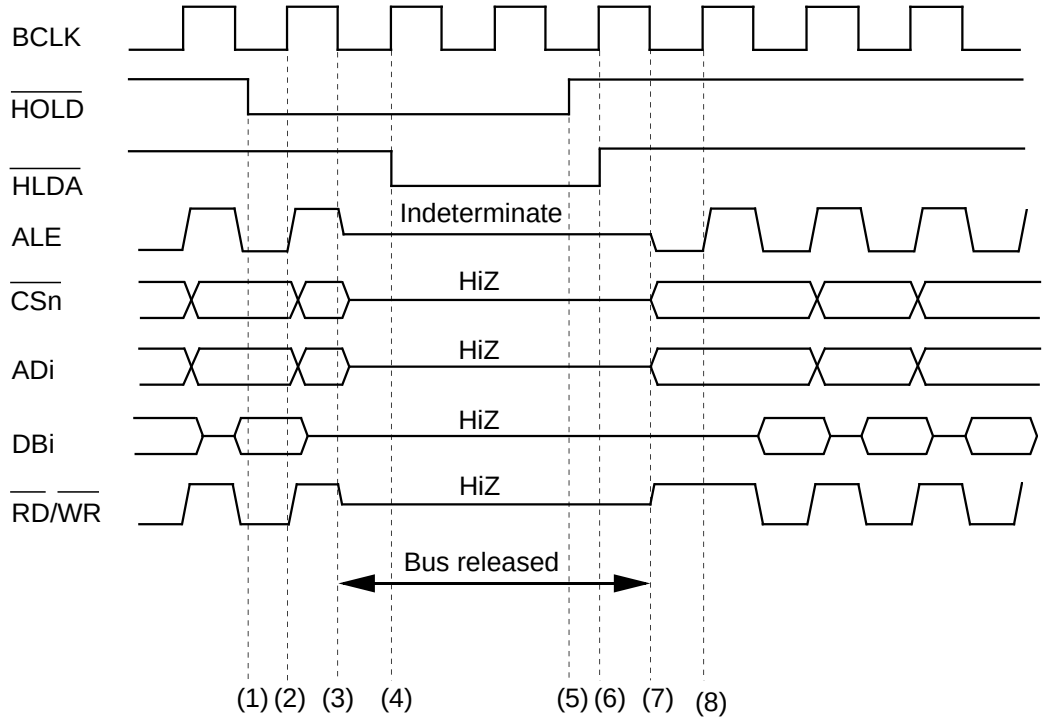
  An internal clock signal having the same phase as BCLK is output.

- **$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$**

  High-impedance state. The case in which ports are selected by the chip selection control register too falls under this category.

Figure.4.5.1 shows an example of relinquishing external buses.

Timing chart



(1) An “L” level is input to the HOLD pin.
(2) HOLD is detected.
(3) The CPU releases the bus.
(4) An “L” is output to the HLDA pin.
(5) An “H” is input to the HOLD pin.
(6) An “H” is output to the HLDA pin.
(7) The CPU not releases the bus.
(8) The CPU resumes using the bus.

Figure 4.5.1. Example of releasing the external bus

4.6 Precautions for External Bus

Description When the MCU enters wait mode while operating in memory expansion mode or microprocessor mode, a pin functioning as part of the address or data bus retains it's state on the bus before wait mode is entered. Shift to single-chip mode and output an arbitrary value in order to reduce current consumption. By shifting to single-chip mode, a pin which was functioning as part of the bus becomes a general-purpose port and can output an arbitrary value. Set the port registers and direction registers after shifting to single-chip mode (this implies that any control pins ($\overline{CS}$, $\overline{WR}$, $\overline{RD}$, etc..) being used for access of an external device be changed as well).

If the port registers and direction registers are set while in memory expansion mode or microprocessor mode, the operation will be ignored.

This is similar when entering stop mode.

Figure 4.6.1 shows the setting procedure to enter wait mode or stop mode.

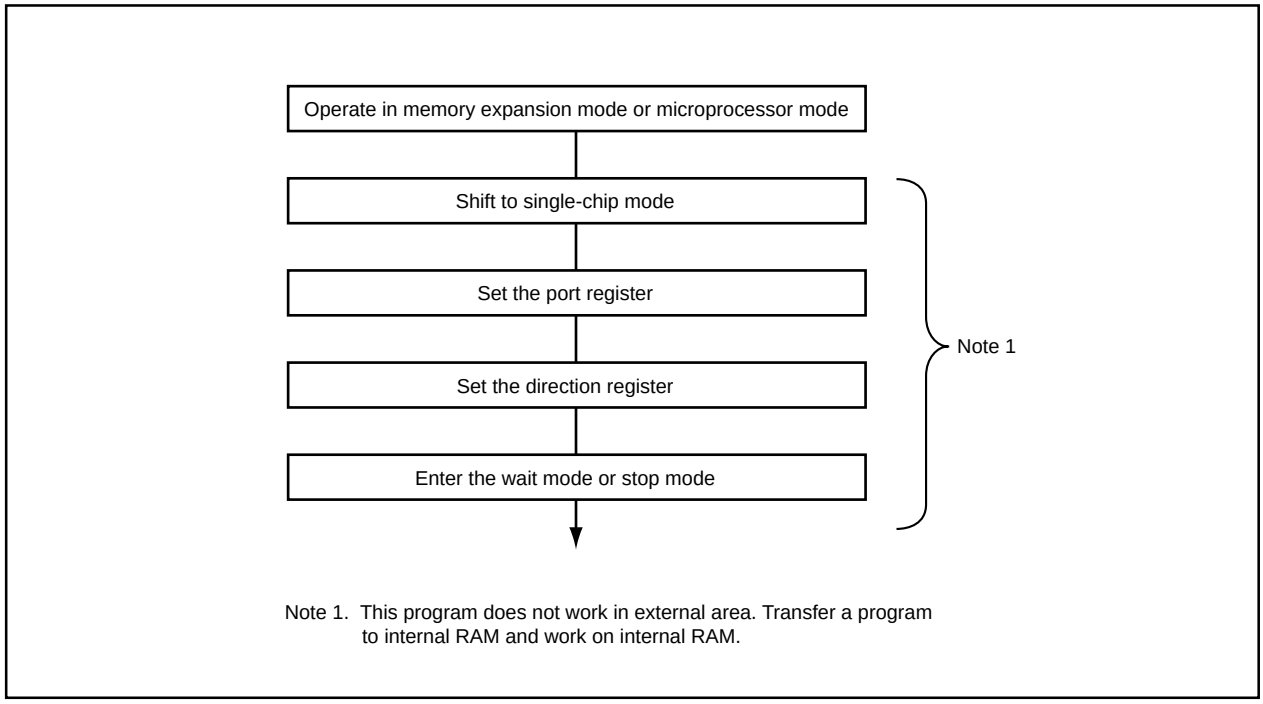

Operate in memory expansion mode or microprocessor mode
Shift to single-chip mode
Set the port register
Set the direction register
Enter the wait mode or stop mode
Note 1
Note 1. This program does not work in external area. Transfer a program to internal RAM and work on internal RAM.

Figure 4.6.1. Setting procedure to enter wait mode or stop mode

THIS PAGE IS BLANK FOR REASONS OF LAYOUT.

Chapter 5

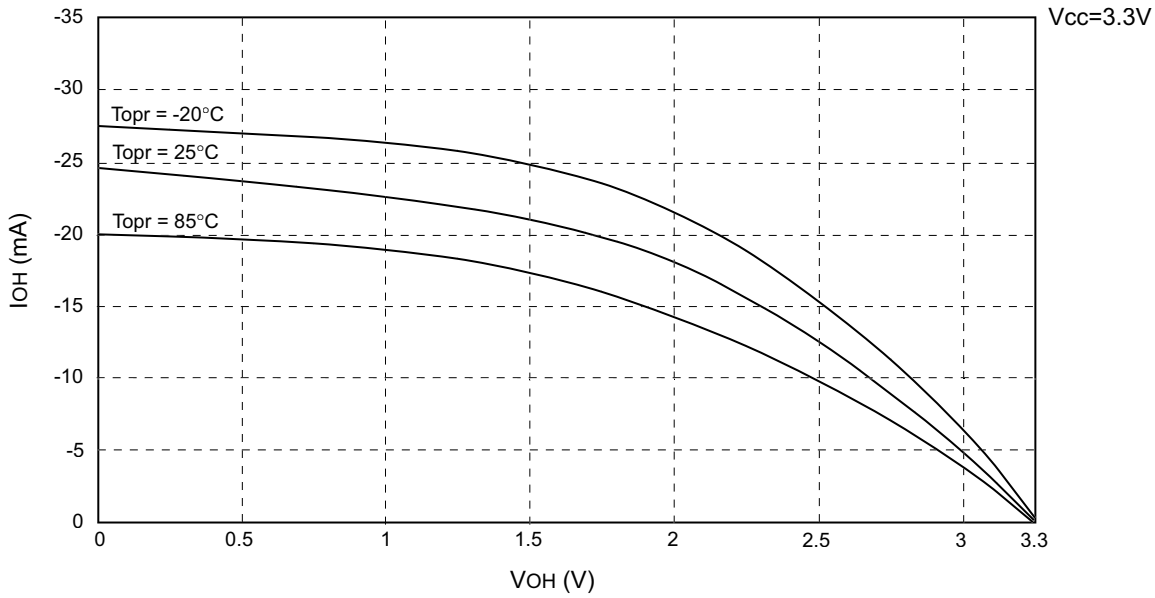
Standard Characteristics

5.1 DC Standard Characteristics

Standard characteristics described in this chapter are just examples and not guaranteed. For rated values, refer to "Electrical Characteristics" of Datasheet.

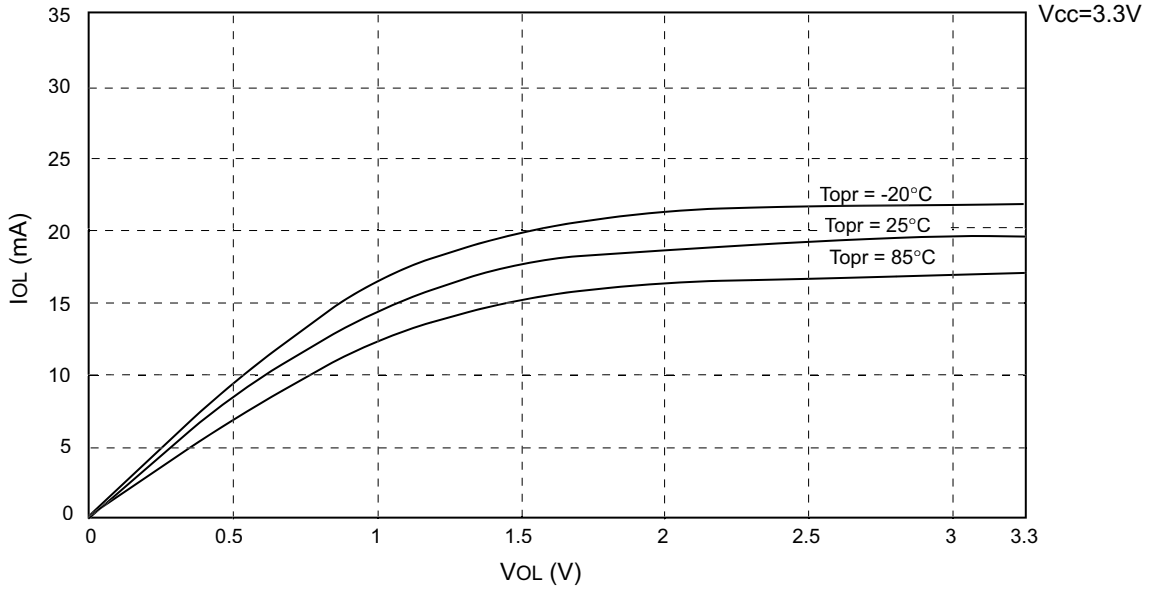
5.1.1 Port Standard Characteristics

Figures 5.1.1 to 5.1.4 show port standard characteristics.



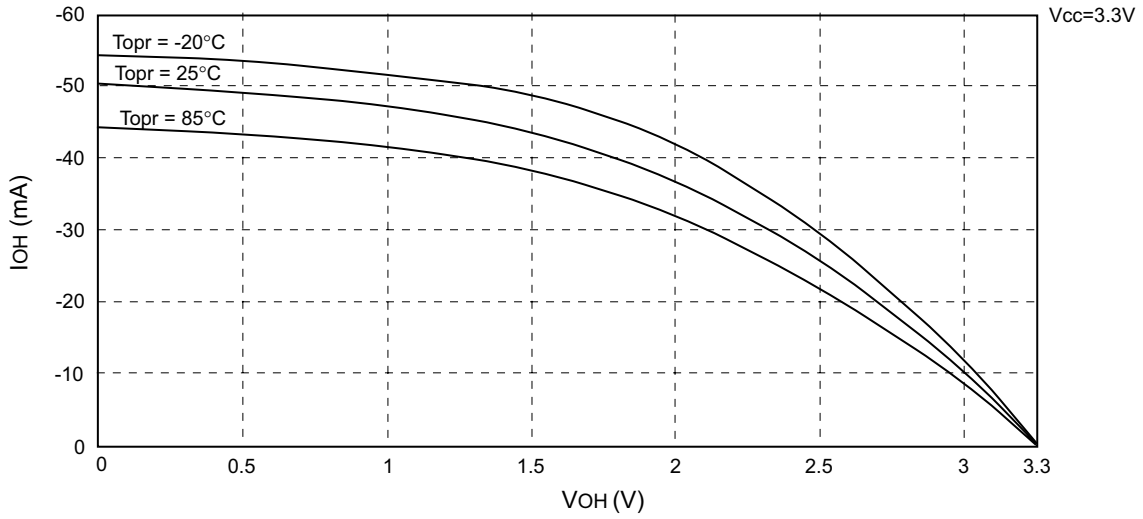
Note 1 : These characteristics are just examples and not guaranteed. For rated values, refer to “Electrical Characteristics” of Datasheet.

Figure 5.1.1. VOH-IOH standard characteristics of ports P0 to P10 (except P63, P67, and P85)



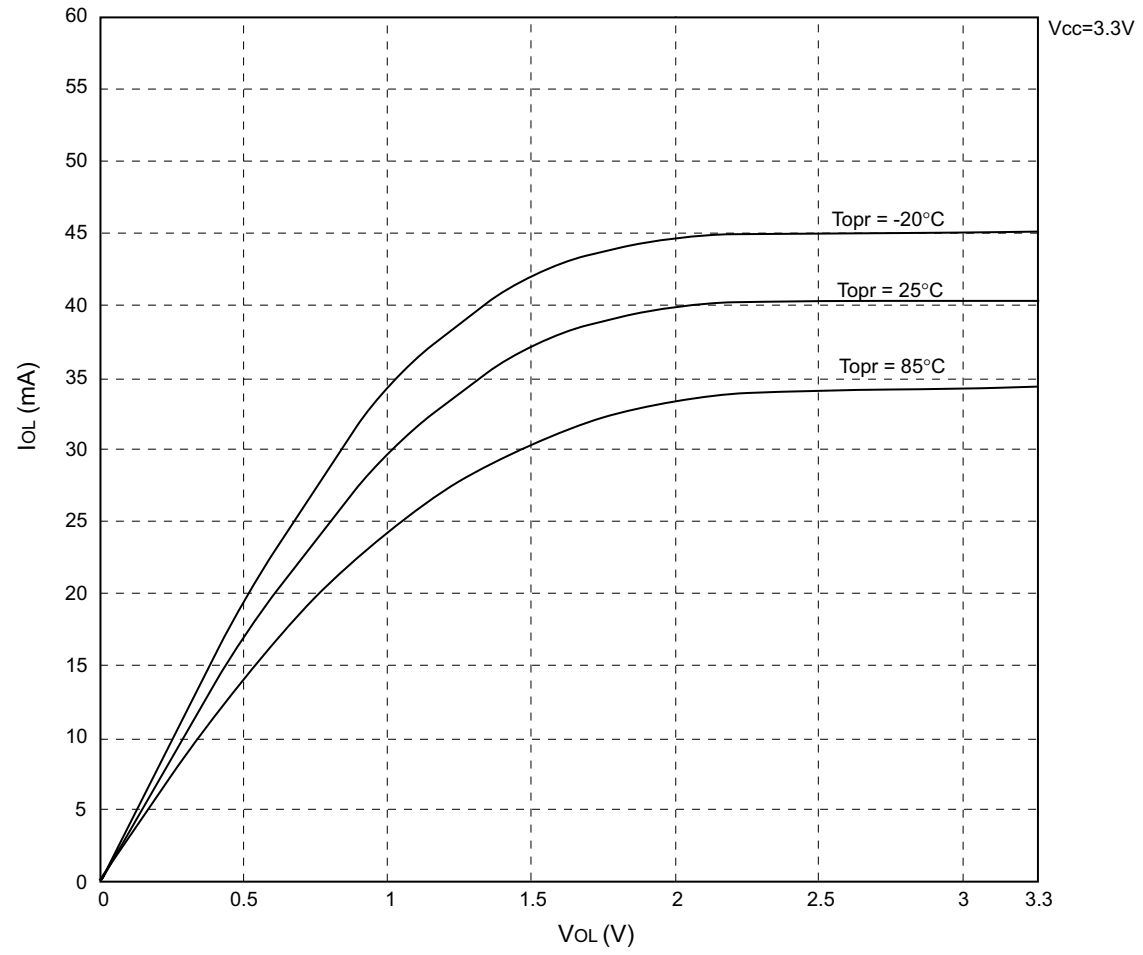
Note 1 : These characteristics are just examples and not guaranteed. For rated values, refer to “Electrical Characteristics” of Datasheet.

Figure 5.1.2. VOL-IOL standard characteristics of ports P0 to P10 (except P63, P67, and P85)



Note 1 : These characteristics are just examples and not guaranteed. For rated values, refer to “Electrical Characteristics” of Datasheet.

Figure 5.1.3. V_{OH}-I_{OH} standard characteristics of ports $P6_3$ to $P6_7$



Note 1 : These characteristics are just examples and not guaranteed. For rated values, refer to “Electrical Characteristics” of Datasheet.

Figure 5.1.4. V_{OL}-I_{OL} standard characteristics of ports $P6_3$, $P6_7$, and P_7 (when high drive selected)

5.1.2 Vcc-Icc Characteristics

Figure 5.1.5 and Figure 5.1.6 show Vcc-Icc characteristics.

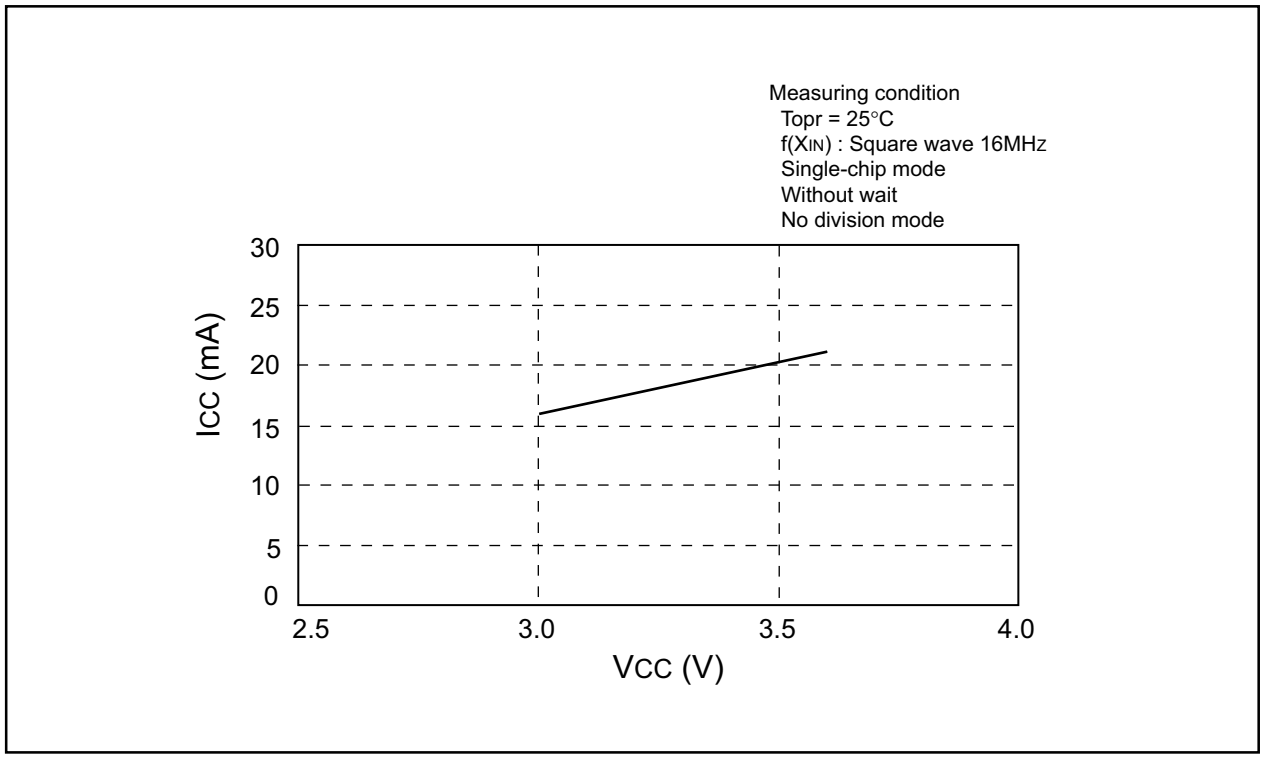


Figure 5.1.5. Vcc-Icc characteristics (Mask version)

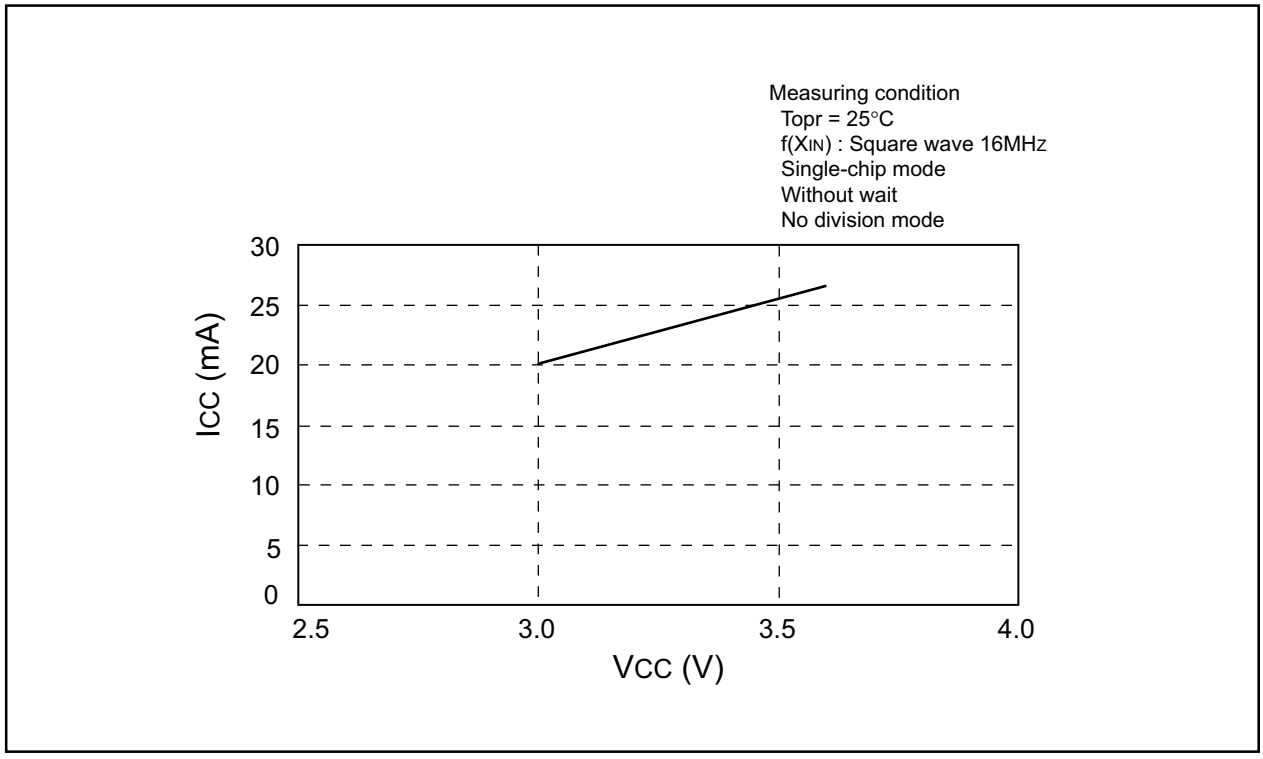


Figure 5.1.6. Vcc-Icc characteristics (Flash version)

REVISION HISTORY	M30245 Group User's Manual

Rev.	Date	Description	
		Page	Summary
A	Jan 24, 2003	–	First edition issued
2.00	Oct 16, 2006	6	2.2.1 (c) "3 types" → "2 types", "an external input signal" deleted
		10	Figure 2.2.4 UDF: bit 5-7; R "○" → "–"
		11	Figure 2.2.5 ONSF Note 3 added
		28	2.2.10 deleted
		37	Figure 2.2.34 deleted
		39	2.3.1 (3) "Error detection" revised
		42	Figure 2.3.2 UiRB Note 1 deleted, ABT; W "×" → "○"
		43	Figure 2.3.3 UiMR; When reset revised, Note 3 added
		44	Figure 2.3.4 UiC0: bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised, Note 2 revised, Note 4 added UiCi Note 1 added
		47	Figure 2.3.6 UiC0: bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised
		50	Figure 2.3.8 Example of operation revised
		51	Figure 2.3.9 UiMR: bit 0-2 "Set the corresponding ..... to "0" when receiving." → "Set the RxDi .... to "0 " when receiving." revised UiC0: bit5; "TxDi" → "TxDi/SDAi and SCLi" revised
		54	Reception (3) revised
		56	Table 2.4.3 Overrun error Description; revised
		57	(c) LSB/MSB first select function added
		60	Figure 2.4.3 UiRB Note 1 deleted, ABT; W "×" → "○"
		61	Figure 2.4.4 UiMR; When reset revised, Note 3 added
		62	Figure 2.4.5 UiC0: bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised, Note 2 revised, Note 4 added UiCi Note 1 added
		66	Figure 2.4.8 UiC0: bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised
		70	Figure 2.4.11 UiMR Note 1 added, UiC0, bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised
		72	2.4.4 added
		74	Figure 2.4.13 Example of operation (when direct format) revised, Note 2 added
		75	Figure 2.4.14 UiC0: bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised
		79	Figure 2.4.17 UiMR Note 1 added, UiC0, bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised
		88	Figure 2.5.2 UiRB Note 1 deleted, ABT; W "×" → "○"
		89	Figure 2.5.3 UiMR; When reset revised, Note 3 added
		90	Figure 2.5.4 UiC0: bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised, Note 2 revised, Note 4 added UiCi Note 1 added
		91	Figure 2.5.5 UiSMR: bit 7 revised, UiSMR2: bit 7 revised, Note 1 added
		92	Figure 2.5.6 UiSMR3: bit 5-7 revised, Note 4 "The amount of delay varies with the load on SCLi and SDAi pins." added
		93	Figure 2.5.7 Note 2 added
		96	Figure 2.5.9 UiC0: bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised
		100	Figure 2.5.12 UiMR Note 1 added, UiC0: bit5; "TxDi" → "TxDi/SDAi and SCLi" revised
		104	Figure 2.5.15 UiC0: bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised
		108	Figure 2.5.18 UiMR Note 1 added, UiC0: bit 5; "TxDi" → "TxDi/SDAi and SCLi" revised
		110-120	2.6 added

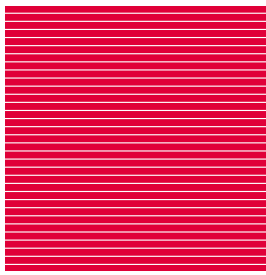
Rev.	Date	Description	
		Page	Summary
2.00	Oct 16, 2006	122	Figure 2.7.3 FSC: bit 2, 1 revised
		124	Table 2.7.1 revised
		125	Table 2.7.3 revised
		138	2.8.2 •USB control register; "the minimum 250ns of delay" → "a minimum 187.5 ns of delay (three cycles of BCLK)"
		143	(2) Enable of USB Function Control Unit; 3,7,8 revised
		144	Figure 2.8.16 "Wait for 4 or more cycles of ϕ" deleted
		145	Figure 2.8.17 FSC: bit 2, 1 revised
		146	Figure 2.8.18 revised
		154	(2) USB Endpoint 0 Interrupt: "• The SETUP_END flag ....." added
		155	(3) USB Function Interrupt: "• The last ACK for control rea ...." added
		157	• Artificial SOF Function revised
		163	• USB Suspend Mode Control: 1 revised and 5, 6, Note added
		165	-Returning Routine of USB Function Control Unit: 5 deleted, "(other than the FSC)" → "(other than the USBC,USBAD,and frequency synthesizer related registers)"
		167	Figure 2.8.29 added
		174	• USB endpoint 0 MAXP register: "SET_DESCRIPTOR" → "GET_DESCRIPTOR"
		210	2.8.9 "(1) USB Communication" added, (2) Peripheral Circuit; "between the USB D+pin and the USB D-pin or" deleted
		211	Figure 2.8.51 revised
		212	(3) Register, Bit: "(addresses 0300_{16} to $033C_{16}$, excepting FSC)" → "(other than the USBC,USBAD,and frequency synthesizer related registers)"
		250	Figure 2.11.2 CRCMR, CRCSAR; When reset revised CRCSAR Note added
		252	2.11.3 "The target SFRs include the .... the UART-related registers." → "The target SFRs include .... Serial Sound Interface-related registers."
		258	2.13.1 "When the external bus .... to the external areas." added
		274	2.16.1 (2) "Reducing the driving capacity .... in power consumption." deleted
		276	(3) revised, (4) added
		280	Figure 2.16.5 (3); "following JMP.B instruction" added
		281	Figure 2.16.6 "Insert at least four NOPs after the WAIT instruction is executed." → "Insert JMP.B instruction before .... NOPs after the WAIT instruction."
		282	2.16.4 (3) revised, (4) added
		283	(d) deleted
		284	Table 2.17.1 revised
		286	Table 2.17.3 AV_{SS}, V_{REF}, USB D+, USB D-, LPF, VbusDTCT, Note 6 added
		287	Figure 2.17.2 PCR revised
		289	Figure 2.17.3 PD9; When reset revised
		303	3.3 deleted
		320	Figure 3.7.4 "The DMA transfer request from .... is transmit request state." → "The DMA tansfer request from .... in transmit request state."
		327	Figure 3.9.3 revised
		329	Chapter 4 added
		351	Chapter 5 added

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER
USER'S MANUAL
M30245 Group

Publication Data : **Rev.A Jan 24, 2003**
Rev.2.00 Oct 16, 2006
Published by : **Sales Strategic Planning Div.**
Renesas Technology Corp.

© 2006. Renesas Technology Corp., All rights reserved. Printed in Japan.

M30245 Group
User's Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan